Structuration automatique de flux vidéos de télévision

Xavier Naturel

N° d'ordre: 3532

THÈSE

Présentée

devant l'Université de Rennes 1

pour obtenir

le grade de : DOCTEUR DE L'UNIVERSITÉ DE RENNES 1
Mention INFORMATIQUE

par

Xavier NATUREL

Équipe d'accueil : Équipe Texmex - IRISA
École Doctorale : Matisse
Composante universitaire : IFSIC, UNIVERSITÉ DE RENNES 1

Titre de la thèse :

Structuration automatique de flux vidéos de télévision

soutenue le 17 avril 2007 devant la commission d'examen

M. :	Patrick	BOUTHEMY	Président
MM. :	Philippe	JOLY	Rapporteurs
	Bernard	MÉRIALDO	
MM. :	Jean	CARRIVE	Examinateurs
	Lionel	OISEL	
	Patrick	GROS	Directeur de thèse

Remerciements

Je tiens particulièrement à remercier Patrick Gros, mon directeur de thèse, de m'avoir permis d'effectuer cette thèse dans une ambiance particulièrement agréable et propice à l'interaction avec des personnes de cultures scientifiques différentes. Son enthousiame communicatif, sa vision particulièrement claire des problèmes les plus intriqués, et sa capacité à rendre effectives des coopérations multi-disciplinaires ont clairement été à l'origine de ma motivation. Je lui suis particulièrement reconnaissant d'avoir su (ré)-orienter mon sujet en début de thèse vers un domaine qui s'est révélé être encore plus passionnant que prévu. Je lui suis aussi extrêmement reconnaissant d'avoir développé de manière importante le travail autour de la plateforme vidéo de l'équipe Texmex, sans laquelle ce travail n'aurait certainement pas atteint la même ampleur.

J'aimerais remercier l'ensemble des membres du jury pour avoir accepté d'y participer. Bernard Mérialdo, professeur à l'Eurecom, et Philippe Joly, maître de conférence à l'université de Toulouse d'avoir accepté la charge de rapporteur. Je remercie également Jean Carrive, chercheur à l'INA, et Lionel Oisel, ingénieur de recherche à Thomson, d'avoir bien voulu se pencher sur mes travaux de thèse et d'y avoir montré un grand intérêt. Je remercie aussi Patrick Bouthemy, directeur de recherche à l'IRISA d'avoir accepté de présider le jury et d'avoir montré, à plusieurs reprises pendant la durée de ma thèse, de l'intérêt pour mes travaux.

Cette thèse n'aurait sans doute pas pu se faire sans le travail considérable effectué par les ingénieurs chargés de la plateforme vidéo. En premier lieu Cédric Dufouil, à qui je dois le développement de Navitex, et les innombrables services que ce dernier m'a rendu. Je dois aussi à Cédric de longues heures de discussion, de correction de bugs, et de développements qui ont considérablement facilité mon travail. La présence d'Arnaud Dupuis m'a aussi clairement permis de gagner en confort, et de m'abstraire de tous ces problèmes techniques parfois pénibles. Qu'ils soient tous deux remerciés.

Certaines personnes m'ont beaucoup aidé à avancer. Guillaume Gravier, de l'équipe METISS de l'IRISA, est de ceux là. De par son ouverture d'esprit et son degré d'implication au sein de Texmex, il a su me fournir des solutions, et élargir ma réflexion et mes connaissances. Je suis aussi extrêmement redevable à Sid-Ahmed Berrani, pour avoir guidé mes premiers pas en tant que doctorant, pour de longues discussions révélatrices, pour avoir repris le flambeau sur la structuration de flux, et enfin pour ne pas perdre une occasion de me rappeler l'épisode incluant un certain ruisseau de forêt. Merci Sid !

Je souhaiterais aussi remercier Ichiro Ide, pour l'intérêt régulier qu'il a porté à mes travaux de thèse et à sa cordialité lors de ses visites à l'IRISA, ainsi que Julien Pinquier,

qui a toujours eu une oreile attentive et des conseils avisés.

Enfin, je dois aussi quelquechose à tous mes compagnons ou voisins de bureau à l'IRISA, pour leurs discussions, leurs questions, et l'ambiance agréable et détendue qui règne au sein de l'équipe Texmex. Dans le désordre : Claire-Hélène Demarty, Philippe Daubias, Zied Jemai, Brigitte Fauvet, Boris Rousseau, Manolis Delakis, Siwar Baghdadi, Hervé Jégou, François Tonnin, Ewa Kijak... et l'ensemble de l'équipe Texmex.

Je remercie enfin particulièrement les stagiaires qui ont travaillé avec moi, Guillaume Chesnel, qui a su prendre en main le détecteur de texte, ainsi que Vincent Guillemot et Lian Liu, qui ont eu la lourde tâche de faire la vérité terrain.

Finalement, j'aimerais remercier ma soeur Corinne, et mes parents, pour tout ce qu'ils ont pu m'apporter au cours de ces trois ans.

Table des matières

Table des figures

Liste des Algorithmes

Introduction

Nous avons assisté depuis quelques années à un accroissement impressionnant du nombre de chaînes de télévision, tant en France qu'à l'étranger. Cette démocratisation de la production et de la diffusion audiovisuelle engendre des problèmes considérables pour les organismes chargés de l'archivage du patrimoine audiovisuel ou du contrôle de l'application des lois sur l'audiovisuel. En France, l'Institut national de l'audiovisuel (INA) est chargé de l'archivage, tandis que le Conseil supérieur de l'audiovisuel (CSA) est chargé du contrôle. En quelques années, ces instituts ont dû passer de quelques chaînes hertziennes à plus d'une centaine de chaînes à archiver ou contrôler. À titre d'exemple, le fond de l'INA augmente de 540.000 heures par an en moyenne[1].

Deux des principales missions de l'INA sont la sauvegarde du patrimoine national audiovisuel français, et la mise à disposition et l'exploitation de ce patrimoine. La première mission relève du **dépôt légal**, tandis que la deuxième mission comprend, à la fois, l'accès aux archives du dépôt légal, ainsi qu'une exploitation des archives à vocation commerciale. Ces missions se traduisent essentiellement par deux tâches : la sauvegarde et la valorisation. La sauvegarde est une opération coûteuse et très technique, qui doit gérer les différents supports de représentation de la vidéo, la restauration, la numérisation, les problèmes de conservation... C'est surtout sur cette première étape indispensable que se sont, tout d'abord, portés les efforts de l'INA.

C'est également aussi une des missions de l'INA que de valoriser ce patrimoine. Les archives sont valorisées lorsqu'elles sont facilement accessibles et qu'il est possible de retrouver l'information qu'elles contiennent par un mécanisme simple. Il est donc nécessaire d'élaborer un mécanisme qui permette de retrouver des archives pertinentes à partir d'une requête formulée par des utilisateurs. On peut distinguer différents types de valorisation en fonction de la vocation des archives. Dans le cadre du dépôt légal, qui est généralement à destination des chercheurs et des étudiants, via l'Inathèque de France, les utilisateurs sont à la recherche d'informations, parfois sans avoir une connaissance préalable de l'information cherchée. Le système peut donc proposer un grand nombre de documents en rapport avec la requête, parmi lesquels l'utilisateur pourra éventuellement sélectionner des documents plus pertinents. Le cas des archives à vocation commerciale est différent. Il s'agit de proposer ou d'identifier précisément un document en rapport avec la requête, et de proposer ce document au client. Les archives à vocation commerciales nécessitent d'être décrites finement, par exemple pour la vente d'extraits aux

[1] En septembre 2006, source : http ://www.ina.fr/inatheque/presentation/collecte.fr.html.

professionnels de la télévision, ou la création de DVD thématiques. Une telle description, réalisée de manière fine, a une immense valeur ajoutée, qui permet d'organiser le contenu et d'apporter des informations qui sont une véritable mise en valeur des archives.

Le contenu doit donc être décrit pour être exploitable, et ceci quelque soit sa vocation (commerciale ou dépôt légal). Plus généralement, le document doit être indexé. On peut définir de manière générale l'indexation d'un document comme étant l'extraction d'informations qui permettent de faciliter l'accès à ce document. La manière de réaliser cette indexation est loin d'être évidente, particulièrement en ce qui concerne les documents vidéos. En fonction de l'application, du mode de recherche d'information, du document, une certaine indexation sera plus ou moins pertinente. Il n'existe donc pas de façon unique d'indexer un document, ni une indexation « générique », qui permette de répondre à toutes les requêtes.

L'une des manières les plus simples et les plus utilisées de rechercher de l'information, quelque soit le type du document, est de le décrire par des **mots-clés**. Il s'agit alors d'annoter le flux audio-visuel avec un certain nombre de mots-clés. Il existe des difficultés intrinsèques à la description d'un document par des mots-clés, à cause de l'ambiguîté et de la subjectivité que peut introduire une telle annotation. Ces difficultés sont encore plus grandes pour la vidéo que pour un document mono-média, comme le texte ou l'image, puisque la vidéo comprend de l'image, du son et du texte, d'où multiplication des possibilités d'ambiguîtés. Au delà de ces problèmes, c'est aussi la temporalité du document vidéo qui pose problème. Une annotation se réfère à une partie localisée du document, il y a donc aussi nécessité d'effectuer une segmentation de ce document ou au moins une localisation temporelle des annotations. L'indexation de documents audiovisuels regroupe donc plusieurs composantes. La segmentation du document en est une, l'annotation en est une autre. D'autres composantes importantes existent, mais ce sont surtout ces deux dernières qui vont nous intéresser.

Si l'on ne considère l'indexation que comme un processus de segmentation et d'annotation du flux, l'un des intérêts du format numérique est la possibilité de pouvoir réaliser cette indexation de manière automatique. A l'INA, l'indexation est réalisée manuellement, par l'intermédiaire de Médiamétrie[2] pour ce qui concerne le repérage des programmes, et par l'INA elle-même pour les descriptions détaillées. Médiamétrie fournit un relevé très précis des programmes effectivement diffusés. Ces relevés sont effectués de manière semi-automatique, mais avec une composante manuelle assez forte. Cette tâche, très pénible et coûteuse, est inenvisageable sur une centaine de chaînes. Il existe donc un réel besoin d'automatisation de l'analyse des contenus télédiffusés, notamment suite au passage de l'extension du dépôt légal aux chaînes du câble et du satellite en 2002, et de la télévision numérique terrestre (TNT) en 2005, soit un total de 52 chaînes[3].

L'intérêt des méthodes automatiques d'analyse vidéo n'est cependant pas de remplacer le savoir-faire des documentalistes. La valeur ajoutée qu'apporte l'INA aux contenus audiovisuels provient de la qualité des annotations créees par les documentalistes spé-

[2]Médiamétrie est une société indépendante dont l'activité principale est la mesure d'audience, mais qui est aussi dans l'obligation de fournir les relevés de diffusion à l'INA et au CSA.

[3]En septembre 2006.

cialisées et du thésaurus développé par l'institut. Il s'agit plutôt de supprimer les tâches pénibles et/ou à faible valeur ajoutée pour permettre une indexation plus facile et rapide, par exemple en proposant une pré-indexation par des outils automatiques. Ces tâches pénibles sont typiquement la localisation temporelle des événements (débuts et fins d'émissions, découpage en chapitres ou scènes, apparition de personnages, etc...), qui sont des tâches tout à fait adaptées à un traitement automatique.

Problématique

Plaçons nous dans le contexte où de nombreux flux de télévision sont produits chaque jour et doivent être documentés. L'une des étapes qui nous parait primordiale est exactement la tâche allouée à Médiamétrie pour les chaînes hertziennes : il s'agit de repérer les débuts et fins d'émissions et de nommer ces dernières. C'est un travail long et répétitif, et donc pénible et coûteux. Il y a donc un fort intérêt, d'une part, à l'automatiser, pour les chaînes qui sont décrites manuellement et, d'autre part, à pouvoir indexer les chaînes qui ne le sont pas actuellement.

Les flux de télévision de longues durées, c'est à dire de l'ordre de la semaine, du mois, voire de l'année, sont des objets complexes. D'une part, par la richesse d'information et la diversité de leur contenu, et d'autre part, par leur structure même. Cette structure est faite pour être facilement compréhensible par un télespectateur afin que l'enchainement des programmes paraisse naturel. De plus, une certaine régularité quotidienne et hebdomadaire est assurée afin que le télespectateur puisse se repérer grâce à la stabilité de la grille des programmes. Si cette structure est aisément perceptible par un « télespectateur lambda[4] », son extraction de manière automatique est, en revanche, difficile. La question alors posée est la suivante : quelle type de représentation peut on extraire du contenu et laquelle adopter pour permettre une navigation et une recherche d'information aisée dans de grands flux de télévision ?

Par analogie avec les documents textuels, on peut distinguer deux types de représentation du contenu :

La table des index, qui se concentre essentiellement sur le contenu, en extrayant de l'information descriptive, ou en détectant des événements. Ceci peut se faire par une simple description du flux, sans notion de localisation temporelle des descriptions. Toutefois pour des documents longs et hétérogènes, la table des index comporte implicitement une notion de localisation temporelle des descriptions, qui peut être limitée à une simple détection d'évènements ponctuels.

La table des matières est elle, au contraire, focalisée sur l'aspect temporel du média. Elle vise à mettre en évidence la structure temporelle et logique du document. Le processus d'extraction de cette structure est appelée la **structuration**. La structure peut être hiérarchique, c'est à dire qu'elle permet d'accéder à différents niveaux de granularité temporelle.

[4]La distinction entre programmes et inter-programmes n'est pas toujours si évidente que ça, en particulier chez les enfants. La possibilité de confusion a d'ailleurs entrainé l'instauration de lois régulant la diffusion des inter-programmes dans la plupart des pays.

Les approches par table des index et table des matières sont complémentaires. La table des index est plus adaptée à une recherche d'information, alors que la structuration permet essentiellement une **navigation** dans le document. La structuration n'est parfois qu'un préalable à une description plus poussée, et peut donc être une première étape dans un processus d'indexation.

Précisons maintenant notre problématique. Il s'agit de déterminer automatiquement les débuts et fins d'émissions, sur des flux de télévision de longue durée. Ceci est typiquement une tâche de structuration, qui vise ici à déterminer la structure d'un flux de télévision au niveau du programme. Par structure au niveau du programme, nous entendons l'enchainement des différents segments au cours de la journée : segments de programmes et segments d'inter-programmes. Ainsi, la structuration n'est pas qu'une simple segmentation, il est aussi essentiel de caractériser un segment par un certain nombre de caractéristiques : son genre, son titre, un résumé, etc...

La structuration n'est pas limitée à l'extraction de la structure au niveau du programme. Elle peut être hiérarchique, c'est à dire qu'elle peut se faire à plusieurs niveaux de granularité temporelle. Un niveau plus fin serait de structurer les programmes, en distinguant leurs différentes phases et leur organisation interne. On peut aussi construire une structuration de niveau encore plus large que le programme, en identifiant par exemple un nouvel habillage de chaine, ou une forte modification de la grille dûe à un évènement particulier (en particulier sportif : Roland-Garros, tour de France, etc...).

C'est essentiellement la structuration au niveau du programme qui nous intéresse ici. Elle peut être considérée comme une tâche de peu de valeur ajoutée car elle ne crée pas de description, le nom des émissions est déjà connu puisque présente dans le guide des programmes de télévision. Elle est, de plus, difficile à réaliser manuellement, car elle nécessite de longs parcours dans la vidéo, avec d'importants risques d'erreurs (manquer un inter-programme court par exemple). Une automatisation de cette tâche a donc un très fort intérêt.

Il convient ici de faire une rapide digression, en se posant la question de l'intérêt de développer une telle technique à l'heure de l'avènement de la télévision numérique. Le problème pourrait en effet être résolu simplement par des métadonnées créés par les chaînes de télévision et indiquant précisément le début de chaque émission. Les standards de métadonnées pour la télévision numérique, tels TV-anytime [TVA02], ont d'ailleurs prévu une telle possibilité. Néanmoins, nous considérons que cette solution n'est pas réaliste, pour plusieurs raisons :

1. l'insertion de telles métadonnées sera toujours manuelle ou semi-manuelle, des erreurs sont donc possibles, et leur production est couteuse.
2. il est très peu probable que les chaînes investissent dans un système qui aidera les téléspectateurs à supprimer les séquences publicitaires, qui sont leur principale source de revenus. Il est donc illusoire de se reposer sur une éventuelle généralisation d'un tel système à l'ensemble des chaînes.

À noter qu'un système permettant de signaler le début exact d'un programme a été développé pour la télévision analogique. Ce système, nommé PDC/VPS [EBU93] et équipant la quasi-totalité des magnétoscopes, est pourtant un échec. En France, seules

les chaînes publiques ne diffusant pas ou peu de publicités l'utilisent[5], ce qui réduit fortement son utilité puisque ces chaînes ont peu de variabilité dans leurs horaires de programmes. Plus généralement, on observe, dans les systèmes actuels de télévision numérique que les métadonnées associées au flux sont extrêmement pauvres, malgré les possibilités techniques. Certes, le secteur est encore récent, mais il est probable que la réticence des chaînes à fournir des métadonnées soit durable, laissant le soin aux diffuseurs de compléter les informations de description afin de proposer de nouveaux services.

Enfin, aucune métadonnée concernant le signal PDC/VPS n'est enregistrée dans la majorité des archives de télévision, le problème de la structuration des archives de l'INA reste donc entier.

Il existe toutefois des métadonnées intéressantes, qui donnent de manière indicative la liste des programmes et, leur horaire de diffusion. Ces métadonnées sont le guide des programmes, qui existe sous plusieurs formes :

- le guide prévisionnel, fourni par les chaînes 2 à 3 semaines avant la diffusion. Ce guide peut être sur support écrit, comme dans les magazines de télévision grand public, ou sous format électronique (sur un site web, par exemple). L'acronyme anglo-saxon EPG, pour Electronic Program Guide, est fréquemment utilisé.
- le guide instantané, est diffusé en même temps que le flux, par l'intermédiaire de la table EIT, qui décrit le programme courant et les programmes à venir. Les informations du guide instantané sont, en général, plus détaillées et plus précises.

Les informations données par le guide des programmes prévisionnel ne sont que des indications : de nombreux programmes sont manquants, erronés, ou intervertis. Les horaires sont tout aussi peu fiables, avec de nombreux retards ou avances par rapport à l'horaire réel. L'annexe F donne une idée du peu de précision du guide des programmes par rapport à la diffusion réelle. De la même façon, le guide instantané est aussi erroné, mais dans une moindre mesure. Toutefois, ce guide n'est disponible que s'il a été enregistré avec le flux, ce qui n'est pas toujours le cas pour des archives.

Applications

La structuration de flux télévisés peut donner lieu à de nombreuses applications. L'une des plus immédiate est une application grand public de type magnétoscope numérique. Deux évolutions technologiques majeures rendent cette application possible : la baisse des prix et la généralisation de disques durs de grande capacité sur différents types d'appareils domestiques multimédias, et la généralisation de la télévision au format numérique (TNT, TV par ADSL, TV par satellite, TV sur mobiles...). Ces deux évolutions rendent possible le stockage de centaines voir milliers d'heures de vidéo sur des appareils grand public, et certains dispositifs permettent déjà d'enregistrer plusieurs chaînes de télévision en simultané, et ceci pendant une semaine en continu[6]. Devant une telle profusion de contenu, il est évident que des fonctionnalités de recherche et de na-

[5] ARTE et France 5

[6] `http://www.promise.tv/`

vigation seront indispensables. Au delà d'un simple outil pour supprimer les plages de publicités, la structuration devient, dans ce contexte, un moyen de trier et d'organiser le contenu vidéo, en proposant un pré-découpage précis du flux télévisé, associé à une description minimale.

Nous avons déjà évoqué en début d'introduction la problématique du contrôle exercé par le CSA, Cet organisme est en particulier chargé de vérifier les temps de diffusion de la publicité, qui est soumise à de fortes contraintes [sdl]. Une structuration automatique du flux télévisé a clairement un intérêt dans ce contexte puisqu'elle fournit l'ensemble des instants d'inter-programmes. Il reste encore néanmoins à différencier les publicités au sein des inter-programmes. Ceci n'est pas vraiment difficile sur les chaînes hertziennes française, puisque les plages de publicités sont encadrées par des jingles spécifiques.

L'application principale reste évidemment l'indexation d'archives de télévision, en particulier pour l'INA, ou pour d'autres (les chaînes de télévision elles-mêmes par exemple). Une structuration automatique au niveau du programme permet en effet, sur les chaines non indexées à ce jour, leur meilleure valorisation, et sur les chaines indexées manuellement, de supprimer cette tâche pénible et de se concenter sur la description plus haut-niveau du flux.

Stratégie générale et organisation de la thèse

La structuration de flux de télévision telle que nous la proposons se divise en deux grandes parties : la **segmentation en programmes** et l'**étiquetage**. Un schéma général de la méthode est donné par la figure 1.

La segmentation en programmes consiste à détecter les instants de début et de fin des programmes. Cette segmentation est effectuée en détectant les inter-programmes, plutôt que les programmes eux-mêmes. Ceci est réalisé par des méthodes assez classiques, issues de la détection des publicités. La segmentation en programmes est expliquée au chapitre 3.

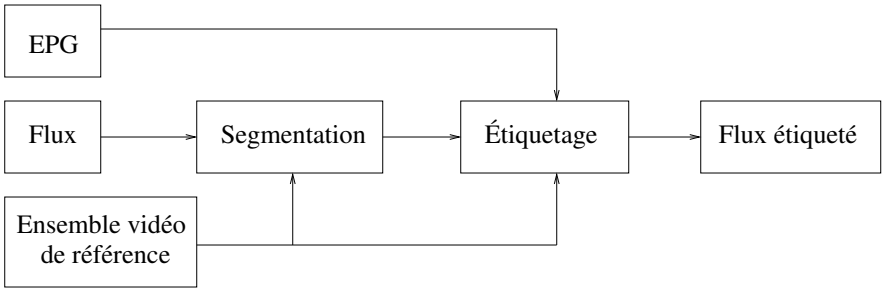

FIG. 1 – Schéma fonctionnel de la méthode de structuration.

La seconde partie consiste à utiliser une information textuelle externe afin d'étiqueter le flux. Nous utilisons pour cela le guide des programmes prévisionnel, qui est une source d'information facilement disponible[7], à défaut d'être exacte, mais qui comporte des

[7] Y compris pour les archives de l'INA, les guides sont conservés.

informations extrêmement utiles pour la structuration. Le nom des programmes y est donné, ainsi qu'un horaire de diffusion approximatif, et éventuellement des informations complémentaires (nom d'acteurs, réalisateurs, etc...). La question est alors de savoir de quelle manière utiliser une telle information. C'est l'opération d'étiquetage, objet du chapitre 4.

La somme des deux processus, segmentation puis étiquetage permet alors, à partir d'un flux vidéo brut, d'en faire ressortir la structure telle qu'un humain la percevrait.

Nous présentons maintenant l'organisation de la thèse en détaillant les contributions de chaque chapitre.

Chapitre 1 Le chapitre 1 présente les travaux antérieurs liés à la structuration vidéo. La structuration de flux de télévision étant une problématique nouvelle, l'état de l'art sur ce sujet est donc limité. Ce chapitre expose principalement l'état de l'art de domaines connexes.

Chapitre 2 Le chapitre 2 présente la méthode de reconnaissance développée pour permettre la recherche des répétitions dans des flux vidéos. La méthode choisie est de type *hachage perceptuel*.

Chapitre 3 Le chapitre 3 explique comment nous effectuons la segmentation d'un flux de télévision en ses différents programmes, à partir de méthodes inspirées de la détection des publicités.

Chapitre 4 Le chapitre 4 propose une méthode d'étiquetage du flux, qui consiste à utiliser l'information présente dans le guide des programmes de télévision, afin d'attacher des informations sémantiques à une segmentation en programmes.

Chapitre 5 Le chapitre 5 traite le problème de la mise à jour de la base de référence utilisée pour détecter les répétitions. Cette mise à jour est essentielle, car l'innovation permanente des flux de télévision fait qu'une base de référence devient vite obsolète, et fait chuter le rappel et la précision de la segmentation en programmes et de l'étiquetage.

Chapitre 6 Le chapitre 6 présente différentes pistes et de directions de travail pour améliorer la structuration de flux de télévision. Ce chapitre expose, entre autres, les travaux effectués en détection et suivi de texte dynamique, des expérimentations sur ce que nous avons appelé l'auto-structuration, ainsi que l'apport des règles pour l'amélioration de l'étiquetage.

Chapitre 1

État de l'art

1.1 Structuration automatique de la vidéo

1.1.1 Segmentation en plans

Historiquement, la toute première tâche de structuration automatique qui a été étudiée est la segmentation d'une vidéo en plans. C'est une tâche très simple au niveau conceptuel puisqu'il s'agit d'identifier les plans[1] produits par le réalisateur. Les frontières des plans sont détectables par une coupure relativement brusque dans le signal vidéo, ce qui est bien adapté à des méthodes automatiques basée sur des mesures de similarité de caractéristiques bas-niveau [TDV00, Han02, Lie01, BGG99]. La segmentation en plans est généralement considérée comme la première étape indispensable vers une structuration de plus haut niveau [Han02] ainsi que pour quasiment tout type d'indexation vidéo. Une segmentation en plans crée, en effet, une structuration très détaillée sur un document vidéo, dont le principal mérite est de permettre de s'affranchir de l'image comme unité temporelle. Le plan devient alors l'unité temporelle de traitement, en le restreignant parfois à une seule *image clé* [FBGS04]. Toutefois, cette unité de structuration est bien trop petite pour que la structure résultante ait un sens pour un humain (à l'exception toutefois d'une analyse cinématographique de la production des plans), le terme de *segmentation* est donc tout à fait approprié.

1.1.2 Macro-segmentation et structuration haut-niveau

Des travaux se sont alors attachés à trouver une structuration de plus haut niveau, essentiellement en essayant de regrouper les plans similaires afin de former des *scènes* [Ven02, AJL95, RHM99]. Une scène est définie comme un ensemble de plans temporellement proches et sémantiquement reliés. Cette tâche est appelée macro-segmentation ou segmentation en scènes. C'est un problème très difficile car la notion de scène n'est pas bien définie. Elle dépend du type d'émission, voire, dans le pire des cas, doit être définie par émission. La détermination de scènes est particulièrement délicate dans les films, où elle obéit à des critères subjectifs. Du fait de ces diffcultés, les résultats sont

[1] Un plan cinématographique est défini comme une prise continue d'images par une caméra.

en général assez mitigés [RHM99, RH04] et il semble difficile d'appliquer ce genre de méthode dans un cas général avec un corpus hétérogène. Toutefois, on peut considérer la segmentation en scènes comme une véritable tâche de structuration, qui permet de déterminer une structure sémantiquement intéressante sur un document, et de permettre, par exemple, une navigation relativement aisée ou un chapitrage automatique. Une des limitations des travaux sur la macro-segmentation, de notre point de vue applicatif, est qu'ils sont réalisés sur des corpus homogènes, c'est à dire constitués d'un seul document (un film, une émission de plateau, etc...). Il n'est alors pas certain que ce genre de méthode soient applicables à la détection de ruptures sémantiques importantes comme un changement d'émission.

Des travaux plus poussés peuvent être menés dans le cadre de ce que l'on appelle les *systèmes spécifiques*. Ces systèmes se concentrent sur des genres de programmes particuliers qui possèdent une structure très forte et peu variable. Ceci concerne essentiellement les journaux télévisés et les retransmissions sportives. À l'aide de connaissances a priori sur le domaine, les systèmes spécifiques sont capables de proposer une structuration de moyen niveau : découpage d'un JT en sujets : reportages/plateaux [EM99, BBP01], découpage en phases de jeu/non-jeu [PBY04] et de haut-niveau : identification des différentes phases de jeu au tennis [Kij03]. Cette structuration est bien plus satisfaisante qu'une macro-segmentation parce que non-équivoque, et permet d'atteindre une structuration véritablement sémantique, d'assez haut niveau, là où la macro-segmentation reste bien souvent au niveau de la similarité visuelle. Ces systèmes utilisent généralement des modèles probabilistes afin de classer les séquences dans différentes classes sémantiques. Les modèles de Markov cachés sont particulièrement populaires [Kij03, EM99, BW98] de par leur capacité à représenter et identifier une séquence d'images (ou de plans). Le problème des systèmes spécifiques, outre leur complexité, est évidemment leur manque de généricité. Une analyse haut-niveau n'est possible que par l'utilisation intensive d'information a priori, et n'est, de fait, pas transposable à tout type de document.

Certains travaux plus ambitieux se sont attaqués à l'extraction de structures véritablement sémantiques dans des documents beaucoup plus variables, comme l'extraction de la trame narrative d'un film [AVBD05], mais ces travaux restent très préliminaires.

1.1.3 Structuration de flux

À notre connaissance, il n'existe que très peu de travaux dont le but avoué est la structuration de très larges documents vidéos, où l'unité de structuration est le programme, à part ceux de Poli [Pol07], développés conjointement avec les nôtres, et les travaux récents de Liang [LLXT05].

Le travail de Haidar [Hai05] s'en approche toutefois assez près. Son but est de définir une mesure de similarité entre deux documents audio-visuels, ces documents pouvant être d'une longueur très importante, et cette similarité mesurant la similarité de *style* entre les documents. L'idée générale consiste à travailler sur un ensemble de séries temporelles mono-dimensionnelles, ces séries représentant les variations de valeurs de descripteurs quelconques. L'approche est donc tout à fait générique, car non dépendante d'un type de vidéo ou d'un descripteur. En particulier, des descripteurs images et audio

sont utilisés simultanément. Haidar donne une application de sa méthode à une tâche de structuration d'une journée de télévision, c'est à dire exactement notre problème. Ses résultats se présentent sous la forme d'une matrice de similarité dans laquelle on voit effectivement la structure du document apparaitre. Toutefois, l'auteur ne dit pas comment extraire cette structure, et cette extraction est loin d'être évidente à partir seulement d'une matrice de similarité.

Le travail de Liang *et al.* [LLXT05] est, à notre connaissance, le seul (avec le nôtre) qui propose une méthode de segmentation d'une vidéo de télévision au niveau du programme directement à partir du flux. Leur but est, néanmoins, un peu moins ambitieux qu'une véritable structuration du flux. Il s'agit pour eux seulement de détecter les débuts et fins de programmes. Les auteurs font remarquer que, d'un jour sur l'autre, les programmes commencent sensiblement à la même heure, et toujours par une séquence de début et de fin identique (le générique). Ils s'appliquent donc à construire un modèle par apprentissage, qui à partir de la détection des génériques identifie un programme par ses heures de début et de fin. Ils exhibent d'excellents résultats au niveau du programme avec des taux de précision et de rappel proche de 100%, mais une exactitude de la segmentation assez faible avec un décalage moyen de 28 secondes entre les débuts/fins réelles et supposés. Cette approche n'est malheureusement pas valable sur des flux plus complexes, en particulier sur la télévision française où on peut difficilement faire l'hypothèse que tous les programmes sont dotés d'un générique. De plus, l'approche modélisation jour par jour n'est valable qu'en semaine, le week-end exhibant des caractéristiques très différentes, et l'approche n'est évidemment pas robuste à un changement de grille. Les auteurs testent sur un corpus de 6 jours de la chaîne Chinoise CCTV-1, mais sur une durée assez limitée de 4 heures par jour, de 17 heures à 21 heures.

Les travaux de Poli [Pol07], développés de façon parallèle aux nôtres, ont effectivement aussi pour but de développer une méthode de stucturation automatique de vidéos de télévision. La grande originalité de son travail est de proposer une méthode de *prédiction* du flux, à partir du guide de la journée courante et d'un apprentissage basé sur une année entière de vérité terrain. La prédiction est réalisée par un modèle de Markov caché, où les états représentent des genres de programmes. Le modèle est modifié pour prendre en compte la durée des états, ce qui n'est pas possible avec un modèle de Markov caché standard, et les probabilités de transition entre états sont dépendantes du contexte. Les résultats obtenus en terme de précision de la prédiction sont très bons, avec des débuts de programmes détectés parfois à quelques secondes près. Les résultats obtenus par Poli montrent, en fait, une très grande stabilité de la grille des programmes, justifiée selon lui par la nécessité des chaînes de fidéliser les spectateurs en leur proposant un programme quotidien stable. Cette méthode souffre, malheureusement, de limitations, notamment lorsque la journée test ne suit pas la grille standard de la chaîne (grille d'été, événement particulier...). Notons la différence importante de cette méthode avec les méthodes de l'état de l'art, qui ont une approche *bottom-up*, c'est à dire qui partent du signal pour en extraire de la sémantique. Ici au contraire, l'approche est *top-down*, dans le sens où l'on part d'une information de haut-niveau, le guide des programmes, pour aller ensuite vers le signal.

Les travaux en structuration de flux télévisés à l'échelle du programme sont, on le

voit, très limités. Il existe, cependant, un problème assez bien étudié qui permet implicitement une structuration d'un flux télévisé en programmes : la détection des publicités. Cette dernière est un moyen de repérer les bornes des programmes et, en conséquence, de faire apparaitre la structure d'un flux de télévision. Bien qu'évidente, cette application des travaux de détection de publicités n'est pratiquement jamais mentionnée, sans doute à cause de la difficulté pratique de la gestion de grand flux vidéo. Nous donnons dans le paragraphe suivant un état de l'art des travaux en matière de détection des publicités à la télévision.

1.2 Détection des publicités

La détection des publicités à la télévision est un sujet qui a potentiellement de nombreuses applications. Satterwhite et Marques [Sat04] identifient deux applications principales : *Commercial tracker* et *Commercial killer*. *Commercial tracker* fait référence à la volonté des entreprises de pouvoir vérifier que leurs publicités sont bien diffusées aux heures souhaitées, le nombre de fois souhaité, ou de surveiller les publicités de leurs compétiteurs. *Commercial killer* fait au contraire référence au souhait des téléspectateurs de pouvoir supprimer les publicités de leurs enregistrements, tout comme les sociétés d'archivage vidéo. Une autre application est de pouvoir indexer l'ensemble des publicités afin, par exemple, de pouvoir les étudier d'un point de vue sociologique ou purement statistique. Enfin, les organismes de contrôle, tels le CSA en France, souhaiteraient vérifier automatiquement les quotas de diffusion de publicité imposés aux chaînes. D'autres applications peuvent encore être imaginées : Covell *et al.* [CBF06] détectent, par exemple, les publicités dans un flux TV, promis à une ré-utilisation (on parle de *repurposing*) afin de les remplacer par des publicités plus récentes ou plus adaptées à l'audience.

Malgré la profusion d'applications, les travaux en détection des publicités ne sont pas très nombreux. Nous reprenons la division proposée Satterwhite et Marques [Sat04] en deux types de méthodes : celles basées attributs et celles basées reconnaissance.

1.2.1 Méthodes basées attributs

Les méthodes basées attributs sont les plus nombreuses et utilisent un ensemble de caractéristiques considérées comme typiques des publicités. La caractéristique la plus populaire est la présence d'images noires qui séparent chaque publicité [LKE97, SMOM02, MD99]. Notons que si ces travaux se réfèrent souvent à des images noires, il s'avère que de façon générale, ces images sont monochromes (noires, bleues, ou blanches, selon les chaînes).

Dans les travaux pionniers de Lienhart [LKE97], les auteurs soulignent que, bien qu'a priori très simple, la détection des images noires est toutefois assez délicate à cause du bruit. Leur détecteur est basé sur le seuillage de la moyenne et de la variance des pixels de luminance. Sadlier et al [SMOM02] travaillent, quant à eux, dans le domaine compressé, et utilisent le coefficient DC de la matrice DCT de luminance d'un bloc 8x8, qui est la valeur moyenne des pixels de ce bloc. La détection des images noires

est alors réalisée par un seuillage des valeurs de ce coefficient DC à partir d'une valeur moyenne de ce coefficient. Le même genre de méthode est utilisé par [MD99], qui souligne qu'une valeur de seuil fixe semble impraticable et ré-ajuste cette valeur de seuil à chaque occurrence d'une image qui dépasse ledit seuil.

Cette détection est néanmoins sujette à de nombreuses fausses alarmes. Pour remédier à cela, la détection est souvent couplée à d'autres caractéristiques comme la présence simultanée de silence [SMOM02], la fréquence des plans [LKE97], la présence de texte [MD99, CYC+05], ou le niveau d'action [LKE97]. L'ensemble de ces éléments sont ensuite regroupés en suivant diverses heuristiques et méthodes ad-hoc, éventuellement en se basant sur les lois en vigueur contraignant la diffusion des publicités, par exemple en Allemagne [LKE97]. Toutes ces méthodes ont en commun l'utilisation de la détection d'images noires pour détecter rapidement des segments possibles de publicité, afin de l'utiliser comme une première étape de localisation de segments candidats.

Une autre méthode possible est de travailler au niveau du plan et de classer chaque plan en tant que publicité ou non, selon des caractéristiques bien choisies. Cette méthode est utilisée soit seule, soit après une étape de sélection des plans à étudier, en général par détection des images noires. Les caractéristiques utilisées, par exemple, par [AFAT04], sont la distribution des plans et la présence du logo de chaîne, avec une classification réalisée par un HMM . D'autres méthodes essayent d'apprendre le genre d'un plan directement à partir de diverses caractéristiques bas-niveau audio et vidéo, en employant des classifieurs évolués, par exemple les SVM [XSH05, DCH04], les réseaux de neurones [Naf94], ou adaBoost contraint temporellement [LQZ04].

Une tendance récente est l'utilisation de caractéristiques audio plus évoluées que la détection de silence. Hua et al [XSH05] appliquent, tout d'abord, un algorithme de détection de transitions audio. Le signal est d'abord découpé en sous-segments avec du recouvrement, puis une mesure de similarité entre coefficients cepstraux et entre l'énergie des deux segments est définie. Le seuillage de cette mesure de similarité permet de détecter les ruptures dans le flux audio. Une classification en parole, musique, silence ou bruit de fond est ensuite réalisée sur chacun des segments produisant un vecteur de dimension 4. Ce vecteur est ensuite concaténé à des vecteurs de caractéristiques images plus traditionnelles (fréquence de changement de plan, nombre d'images noires par seconde, moyenne et variance de la différence pixel à pixel entre images consécutives,...). Toutes ces caractéristiques forment un vecteur de 143 dimensions, qui est utilisé pour une classification par SVM.

Chen et al [CYC+05] utilisent aussi une classification parole/musique mais en tant que traitement de complément d'une méthode par ailleurs classiquement basée sur des heuristiques utilisant la fréquence des changements de plans, les changements de volumes, ainsi que la présence de texte.

À noter que malgré la perception des téléspectateurs d'un accroissement du volume sonore pendant les publicités, cette caractéristique ne peut être considérée comme un indicateur fiable et, à notre connaissance, aucune méthode ne l'utilise. Une étude de l'ENST[2] commandée par le CSA en 2003 montre que le dépassement du volume sonore

[2]École Nationale Supérieure des Télécommunications

des publicités par rapport au volume moyen des programmes n'est pas réellement significatif, bien qu'effectivement existant à l'époque de l'étude. L'étude sur l'énergie du signal audio dans la partie 3.2.2 confirmera que cette caractéristique peut difficilement être utilisée à des fins de classification.

Dans certaines de ces méthodes basées attributs, l'« intelligence » est déportée vers le classifieur, et peu ou pas d'information de contexte ou d'a priori sont utilisés. Plus précisément, les caractéristiques sont concaténées en un vecteur multidimensionnel qui sert à entraîner le classifieur. À charge du classifieur de trouver une surface de séparation entre publicités et non-publicités. Les SVM sont en particulier utilisés à cause de leur capacité à trouver une surface séparatrice non linéaire, mais au vu des caractéristiques utilisées, et le fait que la différence entre publicité et non-publicité est essentiellement sémantique, on peut douter de la capacité d'un classifieur à classer correctement tous les exemples. De fait, et malgré l'utilisation de certains attributs de moyen niveau (en particulier audio), les méthodes basées classification ont les plus faibles résultats, derrière les méthodes à bases d'heuristiques, ou de reconnaissance, étudiées dans le paragraphe suivant (1.2.2).

1.2.2 Méthodes basées reconnaissance

Il existe énormément de travaux qui permettent, à partir d'une vidéo requête, d'identifier des clips vidéos « similaires » dans une base de vidéo. La notion de similarité n'est pas du tout la même selon le domaine applicatif. La recherche par l'exemple, héritée des travaux de recherche en image fixe, est basée sur une similarité globale, généralement à partir de caractéristiques bas niveau comme la couleur, la texture, la forme, ou le mouvement. La notion de similarité est dans ce contexte assez lâche et mal définie : en fonction du (des) descripteur(s) utilisé(s), la similarité sera plutôt une similarité de trajectoire, des couleurs dans la scène, ou une notion plus sémantique comme la présence d'une catégorie d'objets. Les applications sont surtout de l'ordre de la navigation dans une base de vidéos ou de l'identification de clips vidéos « similaires », selon des critères propres à chaque application. Les applications exigeant une base de vidéo importante, de nombreux travaux se sont consacrés à réduire la complexité de la recherche dans une très grande base.

La notion de similarité, telle qu'elle est utilisé en détection de copies, est au contraire bien définie : il s'agit d'identifier une version modifiée d'un document original, soumis à diverses *transformations*. On parle aussi d'*attaques* dans les cas où le but est d'assurer la traçabilité du contenu vidéo, et le contenu peut être soumis à des transformations malveillantes . Ces transformations peuvent être très variées : transformations colorimétriques, géométriques (rotation, translation, redimensionnement...), des insertions, forte compression ou transcodage, tout type de bruit (gaussien, impulsionnel...). La grande diversité des transformations admissibles impose aux méthodes développées d'être particulièrement robustes, mais assez peu de travaux s'intéressent à l'efficacité de la recherche de copies [Ber04, LHÁTJA06, Jol05].

Les deux domaines que nous venons de présenter sont, on le voit, un peu antagonistes, même s'ils peuvent se placer sous la même bannière générale de la recherche

par le contenu. La recherche par similarité possède une notion de similarité assez relâchée et met souvent l'accent sur l'efficacité de la recherche. Au contraire, la détection de copies met l'accent sur la robustesse mais s'intéresse, en général, peu au problème de la complexité de la recherche. Des recherches récentes en détection de copies [Ber04, LHÁTJA06, Jol05], se sont toutefois intéressées avec succès au problème de la complexité de la recherche sur de très grandes masses de données.

Le problème de la détection des publicités à la télévision se place clairement dans une problématique de détection de copies. L'ensemble des transformations admissibles est par contre différent, et se réduit aux transformations apportées par la transmission (bruit gaussien et impulsionnel, changements colorimétriques) par la compression (artefacts de compression divers) et par l'édition (insertions de logos, de bandeaux, redimensionnement géométrique ou temporel). Les transformations sont bien moins sévères que dans la problématique traditionnelle de la détection de copies, mais, par contre, nous ne pouvons pas faire l'économie d'une méthode de recherche efficace, à cause du volume du catalogue de vidéos à parcourir.

Nous présentons dans le paragraphe suivant les méthodes spécialement développées pour la détection des publicités.

1.2.2.1 Méthodes spécifiques à la détection des publicités

Les méthodes basées reconnaissance nécessitent de construire au préalable une base de données de publicités. Afin de décider si un segment vidéo est une publicité, il est comparé aux segments présents dans la base. C'est donc une méthode de recherche par l'exemple dans une base de données de taille et de structure vairiable selon les systèmes. La complexité d'une telle recherche et le fait que les transformations sont assez légères conduisent à choisir des caractéristiques simples. Lienhart *et al.* [LKE97] utilisent un histogramme couleur portant une information spatiale, le *color coherence vector*, et proposent de comparer deux segments en utilisant un algorithme de recherche de sous-séquences. D'autres approches proposent de calculer des signatures à partir de coefficients d'ondelettes [WHHF99], ou à partir du gradient [HB00].

Sanchez *et al.* [SBV02] effectuent une analyse en composantes principales sur les images clés de chaque plan des publicités à reconnaître, ce qui forme un espace de représentation. La méthode de reconnaissance consiste alors à projeter les images-clés des plans à reconnaitre et calculer la distance euclidienne minimale dans cet espace de représentation.

Les méthodes basées plans souffrent généralement du manque de définition d'une mesure de similarité robuste aux déformations temporelles, déformations introduites le plus souvent par l'algorithme de segmentation en plans lui-même. Certains auteurs proposent, toutefois, des approches plus robustes basées sur un algorithme de recherche de sous-séquences [LKE97], la distance d'édition [HZ03], ou ont recours à une recherche exhaustive [PGGM04], au prix d'une complexité supplémentaire. La faible distortion entre deux diffusions d'une même publicité fait toutefois que ces méthodes ont d'excellents taux de précision, le rappel est, quant à lui, conditionné par l'exhaustivité de la base de données utilisée.

Certaines méthodes, que l'on pourrait qualifier de *mixtes*, utilisent à la fois la reconnaissance et des méthodes de classification basées attributs. C'est le cas de Duygulu *et al.* [DCH04] qui détectent des segments de publicités de deux manières indépendantes avant de les fusionner. Ils détectent, d'une part, les répétitions à partir d'une mesure de similarité entre les images clés des plans et, d'autre part, classent les plans en utilisant des caractéristiques audio et couleur. Les résultats des deux détections sont ensuite fusionnés, et sont effectivement meilleurs que pour chaque méthode prise indépendamment.

Une approche différente proposée par Gauch *et al.* [GS06a] est aussi d'utiliser un système mixte, mais d'une manière séquentielle, le but avoué étant d'identifier les nouvelles publicités lorsqu'elles apparaissent, afin de mettre à jour une base de référence de publicités. Les répétitions sont, dans un premier temps, détectées par une méthode de hachage perceptuel [PGGM04], puis sont classées par une méthode de plus proches voisins dans un espace engendré par cinq caractéristiques classiques : le nombre de *coupures* par seconde, le nombre d'images noires aux extrémités du plan, la moyenne et la variance de l'intensité, la moyenne et la variance de la différence entre images adjacentes des moments sur les 3 canaux de couleur.

Notons aussi l'existence de systèmes commerciaux dédiés au suivi des publicités, ou plus généralement de contenu audio-visuel. Citons par exemple NPTV[3], Eloda[4] ou Advestigo[5]. Des méthodes de tatouage sont aussi proposées dans des systèmes commerciaux [Mar] pour vérifier les bonnes diffusions des publicités et autres séquences d'intérêt[6]. Le tatouage possède l'intérêt d'être robuste et d'identifier précisément une séquence vidéo. Son utilisation est toutefois très restrictive puisqu'elle impose d'insérer la marque dans le flux original et sa complexité est un problème pour le passage à l'échelle.

1.2.2.2 Hachage perceptuel

Une approche intéressante est ce que l'on appelle le *hachage perceptuel* (*perceptual hashing*). Ce terme est construit par analogie avec la notion de fonction de hachage en cryptographie. Une fonction de hachage cryptographique prend une entrée de longueur arbitraire et construit une valeur de hachage d'une longueur fixée et de petite taille (quelques centaines de bits). Une comparaison des deux objets est ensuite possible en comparant leur valeur de hachage. Cette technique n'est pas directement transposable pour les documents tels que l'image ou le son puisque deux objets peuvent être significativement différent au niveau bit tout en étant très proches au niveau de la perception humaine. On parle alors de *hachage perceptuel* pour les travaux qui cherchent à construire une fonction de hachage qui renvoie des valeurs de hachage proches pour deux documents audio-visuels similaires. Les applications sont majoritairement dédiées à la robustification des techniques de tatouage [BBH03, FG00] l'identification de contenus et leur éventuelle modification [KP03, MLM04, CS04, OKH02].

[3] `http://www.nptv.fr/`
[4] `http://www.eloda.com`
[5] `http://www.advestigo.com/`
[6] En général, il s'agit de vérfier la non-violation du droit d'auteur.

Le principe de la plupart des méthodes consiste à travailler dans un espace transformé, les transformées les plus utilisées étant la transformée en cosinus discrète (DCT) [CS04, BBH03, KP03, FG00], la transformée de Radon [LCM03, YR04] ou encore le domaine ondelettes [WHHF99, LL00].

La technique de Coskun *et al.* [CS04] consiste à effectuer une DCT 3D sur des séquences de 64 images sous-échantillonnées à une taille de 32x32. Les auteurs extraient ensuite le cube 4x4x4 des coefficients basses fréquences et quantifient les coefficients en les binarisant à partir de leur valeur médiane, pour former une signature de 64 bits. Les distances entre signatures sont ensuite calculées en utilisant la distance de Hamming. Les résultats montrent une bonne robustesse aux transformations de type bruit gaussien additif, flou, compression et, de manière un peu moins flagrante, à des changements de contraste et de luminosité.

Une technique très similaire est utilisée par [BBH03] qui découpe l'image en blocs 128x128, passe dans l'espace transformé DCT et en extrait les 16x16 premiers coefficients avant de les binariser à partir de leur valeur médiane, ce qui forme une signature de 256 bits. Le but est toutefois différent puisqu'il s'agit de prévenir les attaques par copie dans le cadre du tatouage.

La plupart de ces méthodes proposent un moyen de calculer une signature, et fournissent une mesure de similarité, qui est la plupart du temps la distance de Hamming, puisque les signatures formées sont des vecteurs binaires. Elles ne se préoccupent ni de la méthode de recherche, ni de son coût si la base de données considéré est de taille importante. Des travaux intéressants à cet égard sont ceux de Pua *et al.* [PGGM04] et de Oostveen *et al.* [OKH02].

Pua *et al.* construisent une signature à partir des moments jusqu'à l'ordre 3 sur des images couleurs. Ces moments sont ensuite quantifiés et concaténés pour former une signature par image. Les auteurs proposent ensuite de stocker l'ensemble des signatures dans une table de hachage. L'utilisation d'une table de hachage a l'avantage de fournir un ensemble de plans candidats de manière quasi-immédiate. Le problème est alors de décider si ces candidats sont effectivement des répétitions du plan original. Des règles simples de filtrage basées sur la longueur des plans sont utilisées pour réduire le nombre des candidats. Une procédure d'*alignement* des plans qui explore tous les déplacements possibles est ensuite utilisée pour prendre la décision finale. La méthode montre d'assez bons résultats et est capable de travailler en temps réel.

L'approche de Oostveen *et al.* [OKH02] est assez semblable. L'image est préalablement découpée en blocs, sur lesquels sont calculés la valeur moyenne de luminance. La signature est construite à partir de l'ensemble de ces valeurs moyennes, soumises à un filtre spatial et temporel. Le résultat est binarisé par le signe, ce qui au final donne une signature de 32 bits par image. L'intérêt de ce travail est que les auteurs proposent aussi une méthode d'indexation dont le principe est de stocker les signatures dans une table de hachage. Une signature pointe sur une liste des clips mais aussi des positions où elle apparait. Lorsque qu'une requête est effectuée l'ensemble des plans candidats dans chacune des positions candidates sont évaluées. La méthode n'est malheureusement pas évaluée en terme d'efficacité et de rapidité.

1.2.2.3 Autres méthodes

Toutes les méthodes proposées pour mesurer la similarité entre vidéos peuvent a priori être utilisées pour la reconnaissance des publicités. Les transformations entre deux répétitions étant assez légères, le problème de la reconnaissance n'est pas en lui-même un problème difficile. Nous pensons, toutefois, que les méthodes développées pour la recherche par similarité sont intrinsinquement mal adaptées, du fait de leur peu de discrimation, mais aussi de leur faiblesse lorsqu'il s'agit d'effectuer une recherche extrêmement rapidement.

Une approche différente est proposée par Covell *et al.* [CBF06], qui utilisent tout d'abord un algorithme de reconnaissance de segments audio de type hachage perceptuel, inspiré de Ke *et al.* [KHS05], afin de détecter les segments qui se répétent. Les auteurs vérifient ensuite les résultats par une méthode basée image extrêmement simple. Une localisation plus précise des bornes des publicités est ensuite déduite en utilisant tous les segments détectés comme similaires et en forçant leur alignement par une méthode de type Viterbi. D'après les auteurs, cette méthode permet aussi d'isoler, dans l'ensemble des segments qui se répètent, ceux qui appartiennent en fait à un programme, grâce à leur plus grande longueur.

Les travaux de Herley [Her05] sont aussi basés sur l'audio et peuvent s'appliquer indifférement à la radio ou à la bande-son d'une vidéo. Sa problématique est plus générale puisqu'il s'agit de détecter les objets répétés dans un flux audio de grande taille, sans que ces objets ne soient préalablement connus. De la même manière que dans [CBF06], la distinction entre publicités et émission est faite à partir de la longueur des segments détectés comme étant répétitifs.

Les travaux de Joly [Jol05] sont, par contre, particulièrement intéressants puisqu'ils se placent dans une problématique de détection de copies d'extraits vidéos avec la forte contrainte de pouvoir travailler avec une taille de base de référence gigantesque, de l'ordre de 50.000 heures. Il utilise des descripteurs locaux construits à partir des points détectés par un détecteur de Harris. Une telle robustesse n'est pas forcément nécessaire et peut même s'avérer handicapante d'un point de vue applicatif. Elle est toutefois paramétrable, pour revenir à une notion de duplicats et non de copies fortement déformées. La grande force des travaux de Joly est d'être capable de rechercher ces extraits vidéos dans des bases vidéos gigantesques (50.000 heures et plus) tout en conservant une vitesse de recherche suffisante pour faire du monitoring en temps réel différé. Ces travaux sont actuellement les plus performants parmi les méthodes de reconnaissance basées image.

1.3 Discussion

1.3.1 Limitations des méthodes existantes

De nombreuses critiques peuvent être formulées à l'égard des méthodes détection de publicités basées attributs. La première et la plus importante est que ces méthodes ne détectent que les publicités. Or, dans un flux de télévision, de nombreux segments

ne sont ni de la publicité ni un programme : ce sont ce que nous appelons des inter-programmes (IP) : bande-annonce, parrainage, jingle... Cette diversité des segments à détecter fait que la plupart des méthodes exposées précédemment sont moins pertinentes. Les seules méthodes existantes à notre connaissance à faire cette distinction [DJN+02] rapportent toutefois de meilleurs résultats lorsque la notion de « publicité » est étendue à celle d'inter-programme, et [CBF06], qui remarque aussi que les méthodes proposées jusqu'alors ne sont pas forcément adaptées aux IP. Nos observations ont montré que les images noires, qui sont sont utilisées par bon nombre de méthodes, délimitent surtout les publicités, et sont moins ou pas utilisées aux bornes des autres IP. De plus ces séparations par images noires sont inexistantes dans certains pays [LQZ04], et en France, selon les chaînes, ces images sont en fait bleues, blanches ou noires. On devrait donc plutôt parler d'images monochromes.

Les méthodes de classification à partir de caractéristiques bas-niveau auront de grandes difficultés à détecter les bandes annonces ou les jingles car leurs propriétés sont très semblables à celles des programmes. Les méthodes détectant le logo de chaîne ne perdent pas leur pertinence, mais ne peuvent fournir qu'une indication assez peu fiable : les logos sont difficiles à détecter du fait de la présence d'effets de transparence et de mouvement, et leur présence dans un programme n'est pas systématique.

Les méthodes à base de reconnaissance conservent tout leur intérêt pour détecter des IP mais elles sont aussi soumises à de fortes limitations. La principale est le fait qu'elles ne peuvent détecter que des IP connus. Le problème d'une mise à jour dynamique de la base n'est à notre connaissance que très peu traitée. Lienhart [LKE97] le traite de façon sommaire, et l'article ne propose pas d'expérience testant la méthode. Une approche récente et prometteuse est néanmoins donnée par [GS06a].

De plus, les méthodes développées jusqu'à présent accordent peu d'importance à la taille de base de données et à l'efficacité des méthodes de recherche. On peut donc douter de leur efficacité en pratique avec une taille de base conséquente. Les travaux de Joly [Jol05] font exception.

Un autre manque des travaux précents est relatif au faible volume de données sur lesquelles les algorithmes sont testés ou à leur homogénéité. [XSH05, HB00, AFAT04, LQZ04, DCH04] testent sur moins de 20 heures de vidéo, mais surtout sur des corpus homogènes : journaux télévisés, ou films, ou encore corpus construits à la main. Les résultats sont donc biaisés par le fait que les algorithmes ne sont pas évalués dans une situation réelle, où le contenu est hétérogène et imprévisible, et le volume de données fait que de nombreuses situations non standard sont présentes. Les auteurs de [DJN+02] testent toutefois sur un corpus hétérogène mais d'une durée de 8 heures seulement. Quelques travaux sont testés sur de la télévision avec un corpus plus imposant, Liang *et al.* [LLXT05] testent sur en effet 6 jours de télévision, mais malheureusement découpés en période de 4 heures, ce qui réduit le corpus à 24 heures mais surtout possède peu d'hétérogénéité. Covell at al. [CBF06] testent sur 4 jours entiers de télévision, ce qui est un corpus très correct. Encore une fois, Joly [Jol05] fait exception en testant sur des corpus imposants[7] et très hétérogènes, chaînes généralistes françaises, étrangères, et

[7] 50.000 heures, soit 2083 jours, soit presque 6 ans.

archives anciennes.

Un point important est qu'un certain nombre de méthodes de détection de publicités n'utilisent pas le contexte dans lequel sont diffusées ces publicités, c'est à dire que ces méthodes essayent de déduire si un segment est de la publicité à partir de ses caractéristiques propres, sans prendre en compte, par exemple, sa position dans le flux. Cette non-utilisation du contexte rend la tâche beaucoup plus ardue. Une détection des publicités sans utilisation du contexte dans laquelle elle a été diffusée nous semble un peu illusoire.

1.3.2 Synthèse

L'étude qui vient d'être faite nous a montré que très peu de recherches ont été effectuées sur la structuration de flux de télévision au niveau du *programme*. La disponibilité des données, la lourdeur et le coût tant financier que calculatoire de la gestion de très grand flux vidéos ont sans doute handicapé le développement de travaux. Des recherches commencent toutefois à apparaitre [LLXT05, Pol07].

Il existe en revanche de nombreuses méthodes développées dans des buts différents mais qui peuvent s'appliquer à la structuration par programmes. Au premier rang de ces méthodes se trouvent celles développés pour la détection des publicités. Parmi celles-ci, les méthodes s'appliquant à reconnaitre les publicités semblent à la fois les plus efficaces et les plus génériques, capable de facilement élargir le concept de publicité à la notion d'inter-programme, et n'utilisant pas de méthodes ad-hoc spécifiques au pays ou au genre de vidéo. De très nombreux descripteurs et mesures de similarité ont été proposés pour reconnaitre deux segments vidéos mais, en revanche, peu d'importance a été donné aux méthodes d'indexation pour gérer de très grandes bases. Les travaux allant en ce sens sont relativement marginaux. De plus, pratiquement aucune étude n'a été effectuée sur un apprentissage automatique des nouveaux segments de publicités lors de leur apparition, problème fondamental des solutions basées reconnaissance.

En résumé, il apparait que pour développer une méthode de structuration de flux de télévision au niveau du programme, il y a tout d'abord un travail important à effectuer sur la segmentation du flux en segments de programme/inter-programme. Pour cela, les méthodes existantes en détection des publicités doivent être étendue à la notion d'inter-programme, doivent pouvoir gérer de grands volumes de vidéo, et idéalement apprendre automatiquement les nouveaux inter-programmes lors de leur apparition. Elles doivent prouver leur capacité à effectuer une segmentation correcte en programme/inter-programme. Enfin, puisqu'il n'existe pas de recherches allant dans ce sens, les travaux que nous présenterons sur l'étiquetage des programmes, au chapitre 4, seront donc à prendre comme une contribution originale.

Chapitre 2

La détection des répétitions : un outil pour la structuration

2.1 Préliminaires

2.1.1 Problématique

Ce chapitre propose une méthode de reconnaissance de segments vidéos. Plus précisément, c'est une méthode qui permet de reconnaitre qu'un segment est « identique » à un autre. Dans le cas particulier d'un flux continu de vidéo, il s'agit de déterminer quels sont les segments qui se répètent au cours du temps. Nous appellons donc ce problème la *détection des répétitions*.

Définissons plus précisément ce que nous entendons par répétition. Un segment vidéo est une répétition d'un segment original s'il est une version modifiée de l'original par une transformation issue d'un ensemble de transformations admissibles. Cet ensemble se réduit aux transformations apportées par la transmission (bruit gaussien et impulsionnel, changements colorimétriques) par la compression, et par des éditions mineures (insertions de logos, de bandeaux, redimensionnement géométrique ou temporel). Les déformations entre deux répétitions sont considérées comme assez faibles, et excluent toute transformation malveillante ou toute ré-édition importante. Les données sont considérées comme provenant d'une seule source et donc, en particulier, les problèmes de robustesse à différents types de compression ne nous intéressent pas.

L'une des idées importantes qui sous-tend cette thèse est l'importance des répétitions pour la structuration des flux de télévision. Les flux sont en effet extrêmement répétitifs, et la fréquence et l'horaire de ces rediffusions ne sont pas anodins. En particulier, les inter-programmes sont sujets à répétition, de par leur nature même, et on peut se demander si l'étude de leurs répétitions pourrait permettre leur détection.

La fréquence et l'horaire de ces répétitions sont toutefois différents suivant le type d'inter-programmes. Les publicités ont une durée de vie assez longue et ont des schémas de rediffusions variables, les bandes annonces ont au contraire une durée de vie très courte, le parrainage est quant à lui très localisé dans le temps car il est en général attaché à un programme spécifique. Au delà de la détection, les répétitions peuvent

donc aussi être utilisées pour la *caractérisation* des inter-programmes.

Cependant les répétitions sont utiles aussi pour les programmes. Le générique d'un programme, de début ou de fin, ou les jingles qui peuvent apparaitre en cours d'émission pour séparer différentes phases de l'émission sont autant de séquences répétées fréquemment qui permettent de détecter l'émission en question une fois ces génériques/jingles connus.

Les répétitions peuvent avoir une autre application que la structuration. Certains auteurs utilisent les fréquentes répétitions d'images d'actualités pour suivre l'évolution d'un sujet au cours du temps [IMK03, WC06]. Les auteurs de [CN05] font remarquer que les séquences les plus répétées sont aussi les plus importantes (attaques du 11 septembre 2001, ...) et que la répétition d'une séquence véhicule donc une information sémantique.

Il existe donc un nombre important d'applications pour une méthode de détection de répétitions qui soit tolérante à seulement quelques modifications mineures. Pour pouvoir être utilisable aisément, cette méthode se doit cependant d'être rapide, et capable de gérer des volumes de vidéo importants. Ce sont les éléments sur lesquels nous nous sommes concentrés. Le paragraphe suivant présente l'idée générale de la méthode.

2.1.2 Stratégie générale

La tâche de reconnaisance en elle-même n'étant pas difficile, de nombreux types de descripteurs peuvent être proposés. Le problème central est en revanche celui de la complexité. Il n'apparait pas envisageable d'utiliser des descripteurs de grandes dimensions qui sont certes robustes mais posent ensuite le problème de la recherche dans des espaces de grande dimension. Une solution a toutefois été proposée par Joly [Jol05], qui montre que cette approche est possible en utilisant des techniques d'indexation sophistiquées. Une légère limitation de sa méthode, liée à son application de détection de copies vidéos est la précision temporelle. Son approche est, en effet, basée sur des images clés, d'où des taux de reconnaissance faibles pour les segments les plus courts sur lesquels peu (ou pas) d'images clés sont présentes. Il suffirait d'augmenter la fréquence d'échantillonnage des images clés, ce qui conduirait à augmenter assez fortement la complexité du système sachant que des segments aussi courts que 2s existent. De plus, rien ne permet de dire comment détecter les bornes précises d'un segment une fois la détection effectuée.

Pour le problème moins contraignant de détections des répétitions, un certain nombre d'auteurs ont proposés l'utilisation de méthodes de hachage perceptuel[1], qui consiste à construire des *signatures* compactes [OKH02, PGGM04, CN05, Her05, CBF06]. La compacité de ces signatures fait qu'elles sont bien moins robustes et aptes à détecter des similarités, mais leur compacité permet, dans certains cas, d'utiliser une *indexation directe*, c'est à dire qu'une comparaison exacte entre signatures est possible. Il est donc naturel d'utiliser dans ce cas une structure de données qui permette une recherche quasi-immédiate, telle une table de hachage, dont l'utilisation est proposée par certains auteurs [OKH02, PGGM04].

Nous avons aussi adopté une méthode de hachage perceptuel, qui possède l'avantage de proposer une approche conjointe description-recherche. Nous définissons, pour cela,

[1] Parfois appelé *fingerprinting*

dans la section 2.2.3, une signature image très compacte, à fort pouvoir discriminant, et qui permet une indexation directe. Nous expliquons ensuite comment organiser les données dans la section 2.3, nous présentons la méthode de recherche dans la section 2.4, puis les résultats et les limites de la méthode dans la section 2.5.

2.2 Extraction des caractéristiques

2.2.1 Présentation

Cette section présente la phase d'extraction des caractéristiques du flux vidéo.

Nous avons choisi le plan comme unité de détection. La première étape est donc une segmentation du flux en plans, et en parallèle, le calcul d'une signature sur chaque image. Les signatures sont regroupées par plan. Le flux est donc décrit par un ensemble de plans, eux mêmes repésentés comme une liste de signatures. Le plan est alors utilisé comme l'unité de reconnaissance. La méthode de segmentation en plans est donnée dans la section 2.2.2 et la signature est définie dans la section 2.2.3.

Fonction FeatureExtraction(V : **vidéo**) : **liste de plans**
  $S^V = ShotSegmentation(V)$;
  Pour chaque plan S_i^V dans S^V **faire**
    Pour chaque image I_{ik} dans S_i^V **faire**
      $\sigma_k = ComputeSignature(I_{ik})$;
      $S_i^V = S_i^V \cup \sigma_k$; // attache chaque signature à son plan d'origine
    Fin Pour
  Fin Pour
  Retourner S^V ;
Fin

Algorithme 1: Résumé de l'extraction des caractéristiques

Le choix d'utiliser une segmentation en plans est contestable, car il est notoire que les algorithmes de segmentation en plans présentent des carences, et que les effets de transitions progressives et le redécoupage temporel font qu'il est peu probable d'avoir deux fois la même segmentation. Il est donc nécessaire d'effectuer un réalignement temporel et d'être robuste à des découpages erronés, ce qui complique, voir parfois empêche, le processus de reconnaissance. Malgré ces problèmes, nous pensons qu'une segmentation en plans possède d'énormes avantages. Elle fournit, en effet, une unité pour la reconnaisance, qui peut ensuite être manipulée en dehors du flux dont elle est issue, et donc permettre de construire une base de vidéos structurée par plans. Avec l'aide de l'indexation directe, nous allons voir que le problème standard de la recherche d'un plan dans une vidéo de grande taille se ramène à une comparaison entre 2 plans, ce qui est bien plus gérable. L'autre solution généralement proposée [KKM03, YDTX04],

basée sur un découpage à pas constant du flux, nous parait plus difficile à mettre en oeuvre, en particulier lorsqu'il s'agit d'identifier les éléments communs entre deux vidéos de grande taille. La plupart des travaux en la matière supposent, en effet, avoir une requête de petite taille, et surtout d'une taille connue, et il n'est pas clair comment cette approche peut être étendue à la découverte de segments de taille arbitraire. La segmentation en plans nous offre un découpage qui a généralement du sens en ce qui concerne les répétitions, puisque de nombreux inter-programmes ne sont constitués que d'un seul plan : jingle, parrainage, mais aussi génériques. Les répétitions constituées de plusieurs plans ne sont pas problématiques, il suffit d'agglutiner les plans reconnus ensemble, pour former le segment dans son ensemble.

De façon plus pratique, les bornes des inter-programmes sont assez aisément détectable par un algorithme de segmentation en plans. Il y a donc assez peu d'erreurs de reconnaissance liée à une mauvaise segmentation.

En résumé, le processus de détection des répétitions (ou de reconnaissance) n'a besoin que de deux caractéristiques extraites du flux : la segmentation en plans et les signatures de toutes les images des plans extraits. L'algorithme 1 résume ce processus d'extraction des caractéristiques.

2.2.2 Segmentation en plans

2.2.2.1 Principe

L'algorithme de segmentation en plans que nous utilisons est basé sur les travaux de [TDV00] et a été choisi essentiellement pour sa faible complexité. Il n'est ici pas question de choisir une approche complexe à cause du volume de vidéo à traiter, nous travaillons par tranches de 24 heures, sur un corpus de l'ordre d'un mois de vidéo[2], mais aussi parce qu'une segmentation en plan parfaite n'est pas indispensable. Les changements à l'intérieur d'un film ou à l'intérieur d'une émission n'ont aucun intérêt pour nous, seules les bornes des programmes et des inter-programmes sont intéressantes d'un point de vue structuration. Une autre propriété que nous recherchons est une stabilité dans la segmentation : une même séquence vidéo diffusée plusieurs fois devrait avoir la même segmentation en plans, afin de ne pas perturber l'algorithme de reconnaissance.

Nous nous basons, de façon très classique, sur un histogramme de luminance quantifié. L'histogramme est quantifié sur 48 niveaux et lissé par un filtre passe-bas d'ordre 5. La mesure de similarité entre histogrammes est tirée de [Kij03] et est donnée pour deux histogrammes h_1 et h_2 par :

$$d(h_1, h_2) = \frac{\sum_{k=1}^{N_{bin}} \min\left[h_2(k) - h_1(k-1), h_2(k) - h_1(k), h_2(k) - h_1(k+1)\right]}{\sum_{k=1}^{N_{bin}} h_1(k)}$$

avec N_{bin} le nombre de niveaux de quantification de l'histogramme. Cette mesure semble légèrement plus robuste que d'autres distances proposées classiquement, et que nous avons brièvement essayées : la distance L_1, $d(h_1, h_2) = \sum_k |h_1(k) - h_2(k)|$ et le cosinus $d(h_1, h_2) = \frac{\vec{h_1}.\vec{h_2}}{||h_1||||h_2||}$.

[2] Voir la présentation du corpus en annexe A pour plus de détails

Un double seuillage est effectué à partir de la distance entre histogrammes d'images consécutives, le premier avec un seuil élevé, $T_h = 0.8$ qui détecte les coupures les plus évidentes. Un seuil plus faible, $T_l = 0.1$ nous donne une liste de candidats, que nous sélectionnons a posteriori.

Sélection des coupures. Cette sélection est effectuée à partir d'un seuillage adaptatif, inpirée par [TDV00]. L'idée directrice est ici d'éliminer les pics générés par des mouvements rapides dans la scène, ou des mouvements de caméra. On calcule donc une indication de l'activité autour de l'image candidate i dans une fenêtre de longueur fixe w. On calcule pour cela les moyennes à gauche et à droite :

$$m_g = \frac{1}{w} \sum_{k=i-w}^{i-1} d_k \quad m_d = \frac{1}{w} \sum_{k=i+1}^{i+w} d_k$$

avec $d_k = d(h_k, h_{k-1})$ la distance entre histogrammes.

Un premier filtrage est alors réalisé pour ne retenir que les pics qui sont significativement supérieurs au bruit ambiant. Ce filtrage est réalisé par :

$$d_k > \gamma m_g \quad \text{et} \quad d_k > \gamma m_d$$

La décision est finalement prise par

$$\alpha_c d_k + \beta_c \frac{d_k}{m_g + m_d} > T_{coupure}$$

Les seuils sont fixés comme suit : $\gamma = 2.3$, $T_{coupure} = 4$, $\alpha_c = 1$, $\beta_c = 1.5$, et $w = 4$.

Sélection des transitions progressives. Une procédure très similaire est utilisée pour détecter les transitions progressives. Cependant les transitions progressives ayant une largeur, nous utilisons une double moyenne. Les deux moyennes calculées précédemment m_g et m_d caractérisent la transition, tandis que $mext_g$ et $mext_d$, définies par :

$$mext_g = \frac{1}{w + w_1} \sum_{k=i-w-w_1}^{i-w} d_k \quad mext_d = \frac{1}{w + w_1} \sum_{k=i+w}^{i+w+w_1} d_k$$

caractérisent le bruit dans une fenêtre w_1 à l'extérieur de la transition. La décision est prise par le simple détecteur à seuil suivant :

$$\alpha_{tp} \frac{m_g + m_d}{d_k} + \beta_{tp} \left(\frac{d_k}{mext_g} + \frac{d_k}{mext_d} \right) > T_{tp}$$

où les paramètres sont réglés comme suit : $\alpha_{tp} = 1$, $\beta_{tp} = 1$, $T_{tp} = 3$, et $w_1 = 10$.

	TF1_30min		Grand échiquier	
	Précision	Rappel	Précision	Rappel
Coupures	90.2	96.6	95.5	97.1
Transitions progressives	71.2	71.2	82.6	52.6

TAB. 2.1 – Quelques résultats de segmentation en plans.

Filtrage des flashs. Cette méthode est développée spécifiquement pour détecter les changements brutaux d'illumination, tels les flashs ou un passage dans la lumière d'un projecteur. C'est une méthode de post-traitement, qui se déroule donc après qu'une première segmentation en plan ait été déterminée, grâce aux deux méthodes précédentes.

La méthode utilise la signature définie pour la reconnaissance, dont la construction est expliquée plus loin dans le manuscrit, en section 2.2.3. Cette signature est très discriminante, et est à même de détecter si des plans possèdent un contenu visuel proche. L'approche est très simple et consiste juste à calculer la moyenne des distances de Hamming entre les signatures de plans consécutifs. Les plans peuvent être sous-échantillonnés, ou coupés aux extrémités afin d'avoir la même taille. Si cette distance moyenne est en dessous d'un certain seuil, alors les plans sont fusionnés.

2.2.2.2 Évaluation

Nous présentons dans la table 2.1 quelques résultats obtenus avec notre détecteur sur deux séquences : la séquence TF1_30min est une vidéo de 30 minutes composée uniquement d'inter-programmes (publicités, bandes annonces, jingles). La deuxième séquence, qui provient de l'INA, est une émission de Jacques Chancel, *Le grand échiquier*, d'une durée d'environ 2h30. Cette dernière comprend énormément de transitions progressives rapides, ce qui la rend particulièrement difficile. Les résultats sont exprimés en terme de rappel et de précision, suivant la méthode standard de calcul de Trecvid [Sme05]. A noter que ces résultats ont étés calculés avant l'implémentation du filtrage des flashs.

Les résultats de segmentation dépendent évidemment très fortement du corpus. A titre d'information, nous donnons dans la table 2.2 les résultats obtenus sur le corpus de la phase 1 du projet ARGOS[3], qui contient deux types de documents : des journaux télévisés et des documentaires. Les journaux télévisés sont sans grandes difficultés, ils suivent des règles de construction assez strictes, et comportent assez peu de transitions difficiles. Les seuls problèmes peuvent se situer au niveau des reportages avec des mouvements rapides ou des flashs de photographes. Les documentaires sont plus difficiles, avec plus de transitions progressives et des images de moins bonne qualité. Les résultats sont donnés de manière globale, c'est à dire sans distinction entre les coupures et les transitions progressives. Ces résultats sont obtenus en incluant le filtrage des flashs.

Le détecteur proposé est loin d'être parfait et n'est notamment pas à la hauteur en ce qui concerne la détection des fondus trop rapides ou trop longs (cf résultats sur la

[3]http ://www.irit.fr/recherches/SAMOVA/MEMBERS/JOLY/argos/presobjectifs.html

	Précision	Rappel
Corpus INA	96.5	90.3
Corpus SFRS	95.2	75.8
Total corpus ARGOS	96.1	85

TAB. 2.2 – Résultats de segmentation en plans sur le corpus ARGOS, mesure Trec

séquence du *grand échiquier*). Les résultats sont toutefois plus que corrects, et si l'on se compare à l'état de l'art, par exemple les résultats de Trecvid 2005 et ARGOS, les résultats sont bons, voir excellents en ce qui concerne les coupures, et plutôt moyen en ce qui concerne les transitions progressives. Toutefois, le détecteur n'a pas pour objectif de détecter les transitions les plus fines, et il se montre adapté à la tâche souhaitée sur les corpus de télévision. Son intérêt réside dans sa rapidité : la table 2.3 donne la vitesse et le temps de traitement d'une vidéo de 24 heures, en donnant comme référence le temps de décodage vidéo. Les temps de traitement indiqués pour la segmentation comprennent le temps de décodage. Nous dissocions aussi les temps de traitement avec et sans filtrage des flashs, puisque ceux-ci nécessitent le calcul de la signature, ce qui est fait dans un autre but. L'intérêt est ici de montrer que la segmentation en plans en elle même est très légère, et ne ralentit pratiquement pas le processus.

	Vitesse (en images/s)	Temps pour traiter 24h
Décodage	270	2h15
Segmentation (sans filtrage)	230	2h37
Segmentation (avec filtrage)	115	5h13

TAB. 2.3 – Vitesse de traitement de la segmentation en plans.

À noter aussi que le détecteur est robuste aux différents types de vidéos puisqu'il a été testé sur des séquences très hétérogènes en terme de contenu, en utilisant différents codecs, et se comporte de manière satisfaisante, ceci sans avoir changé aucun des paramètres de l'algorithme.

2.2.3 Définition de la signature

Cette section est dédiée à la définition d'une *signature* image, qui est le socle de la méthode de détection des répétitions. L'objectif visé est que les propriétés de cette signature autorisent une indexation directe, et permettent donc une recherche quasi-immédiate d'images répétées. La difficulté est de concevoir une signature qui conjugue un fort pouvoir discriminant, avec cette propriété d'indexation directe, et qui soit robuste aux transformations admissibles définies en introduction de ce chapitre.

Nous proposons de construire une signature à partir des coefficients basses fréquences de la transformée en cosinus discrète (DCT). Sur des images naturelles la DCT possède des propriétés proches de celles, optimales, de la transformée de Karhunen-Loève (voir annexe B). Les coefficients basses fréquences dans l'espace transformé DCT sont donc quasiment décorrélés, et portent l'essentiel de l'information. Il est donc naturel de

construire une signature à partir de ces coefficients, et plusieurs méthodes de hachage perceptuel [CS04, BBH03, KP03, FG00] ont proposé l'utilisation des coefficients DCT pour la construction de signatures. Les approches varient légèrement en ce qui concerne le domaine d'application de la DCT et le procédé de quantification.

Ici, l'image est, en premier lieu, sous-échantillonée jusqu'à atteindre une taille proche du format QCIF (176x144 pixels). La DCT est ensuite appliquée sur le canal de luminance de cette version sous-échantillonée de l'image. L'étape suivante est de sélectionner et de quantifier de manière appropriée les coefficients DCT. Cette étape est cruciale car c'est elle qui permet effectivement de produire une représentation compacte mais aussi de construire une représentation qui va autoriser une indexation directe.

Les coefficients les plus porteurs d'information et les moins sensibles au bruit sont les basses fréquences, la matrice nxn supérieure gauche est donc extraite de la matrice DCT, puis quantifiée. Deux schémas de quantification sont proposés :

Quantification par la valeur médiane Ce schéma est très proche de celui proposé par [CS04], bien que développé indépendamment. Les coefficients sont binarisés selon leur valeur médiane, autrement dit, les coefficients supérieurs à la médiane sont quantifiés à "1" et ceux inférieurs sont quantifiés à "0". Ce procédé revient peu ou prou à coder le signe du coefficient, les coefficients étant distribués de façon à peu près équitable autour de zéro. Prendre la médiane permet cependant de maximiser l'information contenue dans le descripteur puisqu'il est alors d'entropie maximale (équi-répartition des coefficients entre "0" et "1")

Quantification de la signifiance [4] Le signe des coefficients est certes une information robuste mais on peut vouloir garder une information sur leur valeur absolue. Nous proposons donc de coder la *signifiance* des coefficients en utilisant un quantificateur à zone morte (deadzone). Les coeffficients appartenant à la zone morte sont considérés comme non-significatif et codés à "0". Les bornes de la zone morte sont définis par le premier et le troisième quartile, la zone morte est donc l'écart interquartile. Les coefficients à l'extérieur de la zone morte sont codés par "1". La figure 2.1 illustre ce shéma de quantification, où les coefficients DCT sont représentés sur un segment de bornes $[C_{min},C_{max}]$. Cette signature est aussi d'entropie maximale.

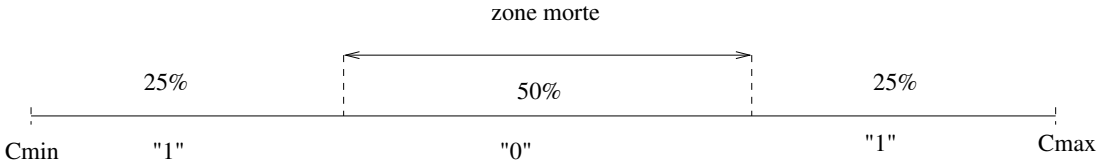

FIG. 2.1 – Schéma de quantification avec zone morte définie par l'écart inter-quartile

Ces deux schémas de quantification produisent une signature de taille n^2. Les valeurs de n considérées sont de 5 à 8, afin d'obtenir une signature de taille inférieure ou égale à

[4] Merci à Hervé Jégou qui m'a suggéré l'idée.

64 bits. Un détail est que le coefficient DC, c'est à dire la moyenne de l'image, n'est pas porteur d'une information discriminante une fois quantifié, puisqu'étant le plus grand, il est toujours quantifié à 1, et ceci quelque soit le schéma de quantification. Nous ne l'utilisons donc pas, et nous le remplaçons par le coefficient en position $n^2 + 1$.

FIG. 2.2 – Image Léna originale

La figure 2.3 donne une interprétation visuelle de la signature construite par le schéma de quantification à partir de la valeur médiane. Les images sont obtenues en effectuant une DCT inverse sur une matrice DCT de taille originale, où la sous-matrice nxn supérieure gauche est quantifiée de la manière que l'on vient de présenter, et où les autres coefficients sont nuls. Il est intéressant de voir l'information conservée par la signature, notamment pour $n = 8$, où l'on commence à deviner l'image originale (figure 2.2), bien que le descripteur ne contienne que 64 bits.

2.2.4 Robustesse de la signature

Les deux procédés de quantification proposés permettent de réduire l'information afin de proposer une description efficace de l'image. Leur but principal est toutefois de permettre une indexation directe, c'est à dire que la signature doit rester invariante aux transformations admissibles. Ceci n'est possible que parce que l'ensemble des transformations admissibles est relativement restreint et que les transformations sont peu sévères.

Afin de vérifier l'invariance supposée de la signature au bruit, nous conduisons une brève expérience, qui consiste à étudier la robustesse de la signature soumise à divers bruits simulés. Les tests présentés ici sont très partiels : ils ne concernent qu'un petit nombre de transformations, et sont limités à une seule séquence. De plus, nous n'avons testé qu'une seule taille de signature. Nous avons choisi la plus grande parmi celles

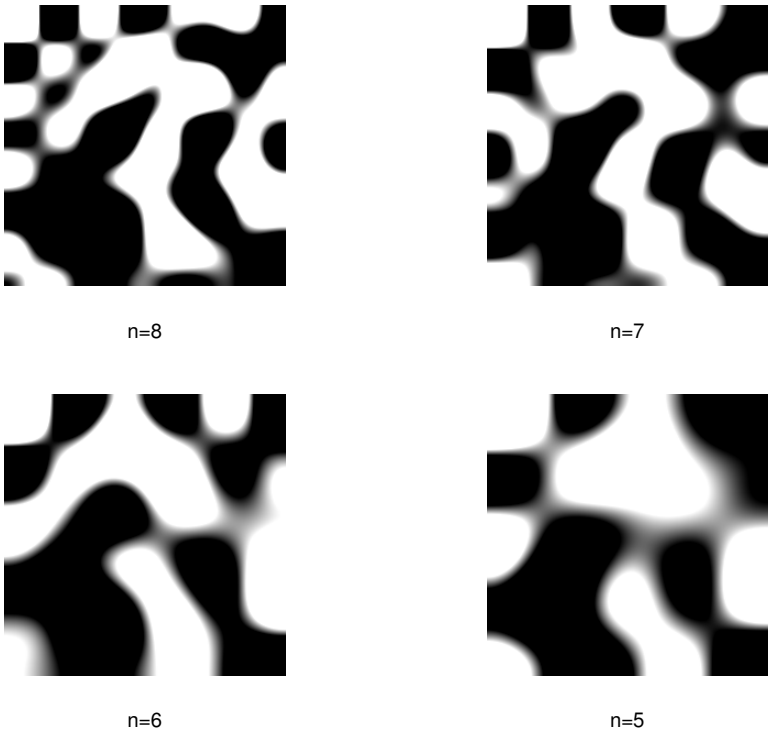

FIG. 2.3 – Reconstruction de l'image Léna à partir de sa signature

considérées (64 bits, $n = 8$), car plus la signature comporte d'information, plus le risque est élevé qu'elle ne respecte pas la propriété d'invariance, ce qui est ce que nous souhaitons tester. Cette expérience est donc surtout utile pour vérifier la possibilité effective d'une indexation directe, et pour avoir une première estimation de l'impact du bruit sur la signature. L'invariance de la signature, et la possibilité d'une indexation directe seront vérifiés plus en détail en section 2.5.1.1, en combinaison avec le mécanisme de recherche, et sur des données réelles.

La robustesse de la signature est évaluée en soumettant une courte séquence de 110 images à divers bruits simulés, considérés comme significatifs des bruits auxquels la signature doit être robuste. Les séquences bruitées sont crées à partir de VirtualDub[5], un éditeur vidéo. Nous appliquons cinq types de bruit avec cet éditeur : un flou gaussien, un changement de luminosité (10%), du bruit uniforme (3%), et enfin de l'insertion de texte (*Texte1* : petite taille, *Texte2* : taille moyenne). Des extraits de séquences bruitées sont montrées sur la figure 2.4. Dans la table 2.4, nous donnons pour chaque séquence bruitée son PSNR et la mesure SSIM de Wang et Bovik [WBSS04]. Le PSNR est défini

[5] http ://www.virtualdub.org/

par :

$$PSNR = 10 \cdot \log_{10}\left(\frac{255^2}{EQM}\right)$$

EQM est l'erreur quadratique moyenne et est définie pour 2 images I_o et I_r de taille mxn comme :

$$EQM = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}||I_o(i,j) - I_r(i,j)||^2$$

Le PSNR donne une mesure de l'importance des modifications au niveau pixels. Les valeurs typiques de PSNR varient entre 30 et 40 dB, pour une image comportant très peu de modifications par rapport à l'originale. Une image sans modifications a un PSNR infini.

FIG. 2.4 – Extrait de vidéos utilisées pour le test de robustesse. de gauche à droite : original, *Texte1*, *Texte2*

Le SSIM, quant à lui, mesure plutôt la modification dans la structure de l'image en calculant un index à partir des statistiques de premier et deuxième ordre de la luminance. Cette mesure est plus apte à quantifier l'impact visuel du point de vue d'un humain. Le SSIM varie entre 0 et 1, une image non modifiée ayant un SSIM de 1. Les deux mesures sont complémentaires, afin d'estimer à la fois les modifications au niveau pixel et au niveau visuel.

Transformation	**PSNR (dB)**	**SSIM**	**sgn identiques (%)**	D_H
Flou gaussien	28.8	0.93	95%	0.38
Luminosité +10%	21.2	0.94	10%	2.5
Texte1	35.5	0.99	79%	0.97
Bruit uniforme 3%	31.6	0.83	76%	0.39
Texte2	25.6	0.97	3%	4.3

TAB. 2.4 – Robustesse de la signature soumise à divers bruits simulés

Afin de mesurer l'impact sur la signature, deux indicateurs sont donnés dans le tableau 2.4 : la moyenne des distances de Hamming entre images de la vidéo bruitée et vidéo originale (D_H), ainsi que le pourcentage de signatures identiques dans la vidéo

bruitée et l'originale. Les tests sont effectués avec une signature de taille $n = 8$, c'est à dire 64 bits.

Les résultats montrent que la signature est robuste à des bruits de type flou, bruit additif, ainsi que dans une moindre mesure à des modifications de contraste. Les incrustations de texte sont, par contre, moins bien supportées, avec seulement 3% de signatures identiques pour la transformations *Texte2*, qui est pourtant une transformation assez peu sévère. Les résultats montrent cependant, que sur cet ensemble de transformations, certes peu sévères, la propriété d'invariance de la signature est vérifiée. En ce qui concerne les distances de Hamming, la signature étant de 64 bits, deux images non corrélées ont en moyenne une distance de Hamming de 32. Les distances de la table 2.4 sont très faibles par rapport à cette valeur, ce qui est encourageant.

En extrapolant ces résultats, on voit, toutefois, que des modifications un peu plus sévères, c'est à dire bandeau+texte inscrusté+bruit, peuvent facilement faire échouer la signature.

```
Fonction ConstructionEVR() : V : vidéo
    T : tableau de plans;
    ht : table de hachage;

    T = featureExtraction(V);
    Pour chaque plan T_i de T faire
        Pour chaque signature σ_ik de T_i faire
            ht(σ_ik) = (i, k);
            // i : index du plan
            // k : position de la signature dans le plan
        Fin Pour
    Fin Pour
    Retourner (T, ht);
Fin
```

Algorithme 2: Construction de l'EVR

2.3 Organisation de l'ensemble de vidéos de référence

2.3.1 Organisation des données

L'idée principale dans l'organisation des données vient du fait de pouvoir utiliser une indexation directe, rendue possible grâce à l'invariance des signatures. L'utilisation d'une **table de hachage** est alors un moyen simple d'obtenir des temps d'accès constants. Nous expliquons maintenant quelles données nous stockons dans la table de hachage et nous définissons ce qui nous sert de base de données, l'ensemble de vidéos de référence (EVR).

Pour chaque image, un couple (i, k) est stocké dans la table de hachage, en utilisant la signature comme clé. i est l'index du plan contenant la signature et k est la position relative de cette signature dans le plan. Les plans sont stockés dans un tableau T. Pour une image caractérisée par un couple (i, k), le plan qui la contient est alors T[i]. On appelle alors ensemble de vidéos de référence (EVR) l'ensemble constitué par la table de hachage et le tableau T. La figure 2.5 résume de façon très schématique l'organisation des données dans l'EVR, et l'algorithme 2 résume le processus de construction de l'EVR dans le cas général.

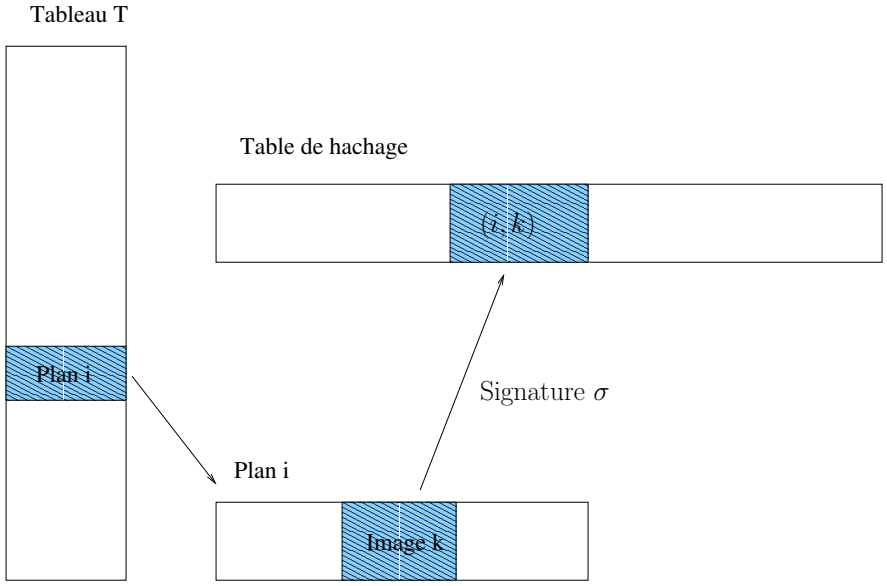

FIG. 2.5 – Schéma de l'organisation de l'EVR

Deux méthodes d'organisation des données sont proposées :

Référence unique une clé pointe sur un seul couple (i, k). Pour une signature donnée, le couple (i, k) est inséré seulement si la signature n'est pas présente dans la table de hachage. C'est cette organisation qui est présentée sur la figure 2.5.

Référence multiple une clé pointe sur plusieurs couples. Pour une signature donnée il y a plusieurs couples candidats. Ces candidats peuvent différer soit par le plan, soit par la position de la signature dans le plan. Ainsi cette méthode peut stocker non seulement plusieurs plans candidats mais aussi plusieurs positions différentes à l'intérieur d'un même plan. En pratique, les éléments de la table de hachage sont donc des vecteurs de couples (identifiant de plan, position de l'image) et non un couple unique.

La méthode par référence multiple est évidemment plus gourmande en mémoire, mais est a priori meilleure, notamment en terme de rappel puisque l'on considère plusieurs candidats par signature. Cette approche est aussi celle adoptée par [OKH02] et [PGGM04]. Oostveen et al. [OKH02] font d'ailleurs remarquer la proximité de cette méthode avec la technique des *fichiers inverses* utilisée en indexation de texte [HFBYL92].

A noter que ces deux méthodes ne font aucune hypothèse sur le type de table de hachage utilisé, et sur le mécanisme de résolution de collisions sous-jacent : les deux méthodes ne diffèrent que par la nature des éléments stockés dans la table. La gestion de la table est extérieure à notre algorithme.

2.3.2 Fonction de hachage

Une table de hachage est une structure de données qui permet l'accès à un élément en un temps constant, autrement dit en $O(1)$. Une table de hachage est constituée d'un tableau et d'une fonction de hachage. Cette fonction assigne un élément de son domaine d'entrée E à un domaine d'arrivée plus réduit, généralement un sous-ensemble de $\mathbb{N}$, $h : E \rightarrow H$, avec $|H| \ll |E|$. En général, h n'est pas une fonction injective, c'est à dire que deux éléments différents de E peuvent être assignés à la même valeur de hachage. Cet évènement est appelé une collision. Le choix de h est crucial pour la performance de la recherche. La conception de h résulte en général d'un compromis entre la réduction du nombre de collisions, et une fonction h de faible complexité. Nous proposons, pour cela, une rapide étude théorique des propriétés d'une fonction de hachage dans le cas idéal où la fonction est de distribution uniforme. Une étude expérimentale est ensuite conduite dans la section 2.3.2.2.

Dans notre cas, le domaine de départ est le domaine des signatures, considéré ici comme un espace binaire de dimension N : $S = \{0,1\}^N$. La signature utilisée en pratique est de 64 bits, ce qui donne une taille théorique de $|S| = 2^{64} \approx 10^{19}$, ce qui est gigantesque. La propriété d'entropie maximale de la signature[6] réduit toutefois l'espace de départ à une taille théorique maximale de $|S| = C_N^{N/2}$, ce qui est toujours gigantesque ($\approx 10^{18}$). En pratique, toutefois, la collection considérée est un sous-espace de S.

Le domaine de hachage est aussi défini comme un espace binaire, de dimension M, $H = \{0,1\}^M$, de taille $|H| = 2^M$. Pour des raisons pratiques de taille mémoire de la table, et pour pouvoir représenter facilement la valeur de hachage sur un entier 32 bits, nous prenons $M = 32$.

La section suivante donne des bornes sur le nombre de collisions N_c dans le cas idéal où la fonction de hachage est de distribution uniforme.

2.3.2.1 Étude de la distribution uniforme

Nous supposons que h est de distribution uniforme. Une collision est définie par l'événement :

$$\exists\,(s_1, s_2) \in S \;\text{ tels que } s_1 \neq s_2 \;\text{ et } h(s_1) = h(s_2)$$

Autrement dit, h n'est pas injective. Nous souhaitons étudier l'existence et le nombre de collisions en fonction du nombre de signatures p et de la taille de l'espace d'arrivée H.

[6] La propriété d'entropie maximale provient du fait que la signature comporte le même nombre de "1" que de "0", le nombre de signatures possibles est alors le nombre de combinaison de $N/2$ éléments parmi N, soit $C_N^{N/2}$

p signatures doivent être rangées dans une table de hachage de taille $|H|$. Si $p > |H|$ le principe des tiroirs[7] fait que la probabilité de collision est de 1. Sinon le classique *paradoxe des anniversaires* donne la probabilité de collision :

$$P_{collision} = 1 - \prod_{k=0}^{p-1} \left(1 - \frac{k}{|H|}\right)$$

En général, il est impossible d'éviter les collisions [FGS90], il est alors intéressant de calculer une estimation du nombre de collisions, qui est donnée par [McK03] :

$$N_c = \frac{C_p^2}{|H|} = \frac{p(p-1)}{2|H|}$$

On peut cependant remarquer que même dans le cas idéal où l'espace d'arrivée est de même taille que l'espace de départ, c'est à dire $p = |H|$, le nombre de collisions reste assez élévé, avec $N_c = \frac{|H|-1}{2}$, alors que h pourrait être bijective, c'est à dire sans collisions. Ainsi même dans le cas où la fonction h est uniforme, le nombre de collisions que l'on peut attendre est très loin du cas idéal déterministe où h serait bijective.

Si l'on souhaite un taux de collision très faible, par exemple $N_c < 1$, le nombre de signatures est borné par $p < \sqrt{2|H|}$. Dans notre cas, $|H| = 2^{32}$, la taille maximale de l'ensemble des signatures $S_{\text{réel}}$ pour que la table soit sans collisions serait de 92681 éléments, soit seulement 2 heures de vidéo[8] !! Il y a donc un compromis à trouver entre le nombre de collisions considéré comme acceptable et la taille de la table de hachage.

Une question intéressante est de déterminer la taille de H qui permet de ne pas dépasser un certain taux de collisions. On souhaite par exemple un taux de collisions inférieur à 1 pour r, c'est à dire $N_c < \frac{p}{r}$, on aboutit à :

$$|H| > \frac{r(p-1)}{2}$$

Ce qui donne la borne minimale pour la taille de la table pour un taux de collision fixé, et un nombre d'éléments à stocker connu.

Inversement, on souhaiterait connaitre le nombre maximal d'éléments que peut contenir $S_{\text{réel}}$ sans dépasser un certain taux de collision, la taille de la table de hachage étant fixée. l'inégalité devient :

$$p < 1 + \frac{2|H|}{r}$$

Dans notre cas, $|H| = 2^{32}$, et on souhaite par exemple un taux de collisions ne dépassant pas 1 pour 100. L'inégalité donne une taille maximale de 86 millions de signatures, soit environ 80 jours de vidéo, ce qui est confortable.

[7] pigeonhole principle en version anglaise.

[8] Il a été observé expérimentalement que d'un segment de N images consécutives, il y a seulement N/2 signatures distinctes

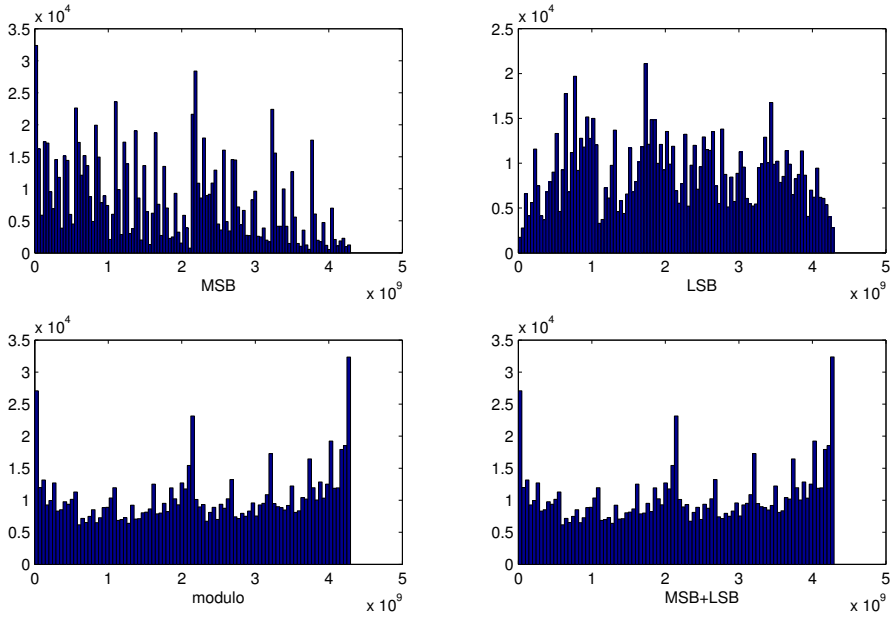

FIG. 2.6 – Distributions des fonctions de hachage proposées

Il est important de remarquer que $N_c \sim \frac{p^2}{|H|}$. Pour espérer un très faible taux de collisions, la table doit être d'une taille comparable à p^2. Si, en pratique, on admet un nombre non négligeable de collisions, on peut cependant prendre une taille de table bien inférieure, de l'ordre de p, par exemple $|H| = 2p$.

2.3.2.2 Choix de la fonction de hachage

Afin de limiter le nombre de collisions, nous devons étudier la distribution des signatures et trouver une fonction de hachage qui la transforme en une distribution uniforme dans l'espace de hachage. Cette fonction doit cependant être de faible complexité, sinon le bénéfice d'avoir peu de collisions est perdu. Nous ne considérons donc ici que des fonctions de hachage qui ne sont que des opérations élémentaires : décalages binaires, sommations. La fonction modulo est aussi donnée à titre de comparaison puisque c'est une fonction de hachage très utilisée.

1. $h(\sigma) = \sigma_{1 \text{ to } 32}$ (MSB - Most Significant Bits)
2. $h(\sigma) = \sigma_{32 \text{ to } 64}$ (LSB - Least Significant Bits)
3. $h(\sigma) = \sigma \mod (2^{31} - 1)$ (modulo)
4. $h(\sigma) = \sigma_{1 \text{ to } 32} + \sigma_{32 \text{ to } 64}$ (LSB+MSB)

Afin de justifier le choix de la fonction de hachage, nous contruisons un histogramme pour visualiser et évaluer la distribution des signatures dans l'espace de hachage. La

règle de Freedman-Diaconis [FD81] est utilisée pour estimer le pas de quantification des histogrammes. La largeur W d'une classe est donnée par :

$$W = \frac{2 * IQR}{\sqrt[3]{N}}$$

où N est le nombre d'échantillons, et IQR est l'écart interquartile. Afin de donner une indication de la proximité de la distribution par rapport à la distribution uniforme, nous donnons la divergence de Kullback-Leibler (KLD). La KLD est définie pour deux distributions p et q comme :

$$d(p, q) = \sum_k p_k \log_2 \frac{p_k}{q_k}$$

La KLD, la variance des classes et le pourcentage de collisions sont donnés dans la table 2.5. La fonction LSB obtient les meilleurs résultats en terme de variance des classes, bien que ce ne soit pas la plus proche de la distribution uniforme au sens de la divergence de Kullback-Leibler. La KLD donne une indication générale sur la forme de la distribution, en particulier ici, elle permet de mesurer si la distribution est « lisse ». La variance mesure, quant à elle, la dispersion, c'est à dire le fait que les échantillons sont distribués autour de la moyenne. La variance est donc plus adaptée pour mesurer la qualité de la distribution d'une fonction de hachage, puisqu'un pic dans l'histogramme sera plus pénalisant pour la variance que pour la KLD, et une classe très peuplée est justement ce que doit éviter une fonction de hachage. Les distributions des différentes fonctions de hachage proposées sont données dans la figure 2.6.

Un exemple de l'impact de la complexité de la fonction de hachage est donné par les fonctions MSB+LSB et modulo. Ces deux fonctions ont des distributions quasi-identiques (indiscernables sur la figure). On peut justifier cela rapidement par le fait que pour un entier n sur 64 bits et $p = 2^{32} - 1$ on a :

$$n = (p + 1) * MSB(n) + LSB(n)$$

et donc

$$n \mod p = MSB(n) + LSB(n)$$

pour $LSB(n) \neq p$, $MSB(n) \neq p$. En pratique la fonction LSB+MSB est pourtant plus rapide, à cause de sa complexité moindre.

Fonction	**KLD**	**Variance**	**Collisions (%)**	**temps de recherche (s)**
Uniforme	0	0	0.01	-
MSB	0.44	6755	42.9	1.34
LSB	0.12	3777	19.1	1.24
Modulo	0.1	4269	2.3	1.47
LSB+MSB	0.1	4269	2.3	1.4

TAB. 2.5 – Propriétés des différentes fonctions de hachage proposées

Les temps de recherche indiqués dans la table 2.5 sont obtenus en prenant pour requête une vidéo de 24h, avec un EVR d'une semaine, soit environ 170 heures. Chaque plan de la vidéo de 24h est recherché dans l'EVR selon le processus décrit dans la section 2.4. Une vidéo de 24 heures comportant en moyenne 20.000 plans, cette requête équivaut à un minimum de 20.000 recherches de plans dans l'EVR.

Les résultats présentés dans cette table semblent montrer que la fonction LSB est la meilleure. Le pourcentage de collision, c'est à dire le pourcentage de signatures qui souffrent effectivement de collision, peut être élevé tant que ces collisions sont équitablement réparties dans les différentes classes. L'indicateur le plus adéquat est donc la variance, ainsi bien sûr que le temps de recherche lui-même, ce qui nous conduit à choisir la fonction LSB comme fonction de hachage. Elle est de plus très facile à implémenter et peu complexe, puisque elle s'implémente sous la forme d'un simple `cast` d'un entier 64 bits vers un entier 32 bits.

2.4 Méthode de détection des répétitions

2.4.1 Définition de l'algorithme

Nous présentons dans cette partie la méthode de détection des répétitions dans sa globalité. La méthode est résumée par l'algorithme 3.

Les premières étapes sont la construction de l'EVR et l'extraction des caractéristiques de la requête, segmentation en plans et calcul des signatures. Ensuite, pour un plan requête donné, l'algorithme teste la présence de toutes les signatures de ce plan dans la table de hachage. Si la table de hachage retourne un résultat, alors le plan candidat est déterminé grâce à l'index stocké dans la table, et une distance entre le plan requête et le plan candidat est calculée. Dès que cette distance est inférieure à un certain seuil la procédure est arrêtée et le plan requête est alors déclaré reconnu. La procédure continue sinon jusqu'à temps que toutes les signatures du plan soient testées. La définition d'une distance entre plans est le sujet de la section suivante.

Il est ici très important de noter qu'une seule signature est suffisante pour trouver un plan candidat. C'est une propriété essentielle de l'algorithme, qui a pour effet de réduire une recherche de la présence d'un plan dans un EVR à un simple accès dans une table de hachage, et ceci quelque soit la taille de l'EVR considéré. En pratique, plusieurs essais peuvent être nécessaires avant de trouver le résultat correct. Ceci est dû au fait que la propriété d'invariance de la signature donnée en 2.2.3 n'est en général pas valable pour toutes les signatures. Le pourcentage de signatures communes entre le plan « original » et le plan « transformé » dépend de la criticité de la transformation, comme montré en section 2.2.4. Ces résultats, bien que partiels, tendent à montrer qu'il y a une très grande probabilité pour que deux plans répétés partagent au moins une signature en commun. C'est l'hypothèse sur laquelle est fondée la méthode de détection des répétitions, et la section 2.5 montrera que cette hypothèse est correcte en général. Ceci explique aussi pourquoi toutes les signatures du plan sont testées tant qu'un résultat positif n'est pas trouvé, cela ne serait pas nécessaire si la propriété d'invariance était toujours vraie.

```
RVD : vidéo ; [EVR]
Q : vidéo ; [requête]
ht : table de hachage ;
D : distance entre plans ;
Seuil : entier ;
T^RVD, T^Q : tableau de plans

(T^RVD, ht) = databaseConstruction(RVD) ;
T^Q = featuresExtraction(Q) ;
Pour chaque plan T_i^Q de T^Q faire
    Pour chaque signature σ_ik de T_i^Q faire
        Si (σ_ik ∈ ht) Alors
            (n, p) = ht(σ_ik) ;
            // n : index dans S^RVD du plan contenant σ_ik, i.e. T_n^RVD
            // p : position de la signature dans T_n^RVD
            Si (D(T_n^RVD, T_i^Q) < TEB_max) Alors
                break ; // les plans sont identiques, on passe au plan requête
                suivant
            Fin Si
        Fin Si
    Fin Pour
Fin Pour
```

Algorithme 3: Algorithme de détection des répétitions entre 2 vidéos

Une différence importante de cet algorithme avec la plupart des algorithmes existants est sa capacité à gérer des requêtes de grandes tailles. En effet, l'approche classique est de considérer un clip vidéo de petite taille, et de rechercher ses occurrences dans une grande base vidéo. Ici, nous considérons des requêtes de grande taille, qui peuvent être de taille équivalente, ou supérieure, à la taille de l'EVR. Cette gestion de grandes requêtes se fait simplement par le biais de la segmentation en plans, un plan devenant alors une sous-requête. Le découpage en plans de la requête comme de l'EVR permet alors d'identifier rapidement les **plans communs** entre deux vidéos. Par abus de langage, on dira que l'on recherche vidéo1 dans vidéo2, ce qui veut dire, en fait, que l'on recherche la présence de chaque plan de vidéo1 dans vidéo2.

Une autre particularité, qui peut se révéler handicapante cette fois-ci, est que le processus ne permet pas d'identifier toutes les occurrences d'une requête dans l'EVR mais seulement sa présence. L'utilisation du hachage renvoie en effet un résultat unique, même si des répétitions de la requête sont présentes en grand nombre dans l'EVR. En fait, le problème est contournable très simplement en inversant requête et EVR. Plus précisément, si on souhaite identifier toutes les occurrences d'un plan P dans une vidéo V de grande taille, alors l'approche intuitive qui consiste à mettre P en requête et V

comme EVR permet seulement d'avoir un résultat sur la présence ou non de P dans V. En revanche, si les rôles sont inversés, alors nous pouvons localiser toutes les occurences de P dans V.

La dernière étape de l'algorithme, et l'une des plus problématique, est la définition d'une distance entre plans. La section suivante expose le problème et propose diverses possibilités.

2.4.2 Distance entre plans - alignement

2.4.2.1 Introduction

De par sa petite taille, la signature définie en 2.2.3 est porteuse de peu d'information. Par conséquent, la signature n'est pas forcément un bon descripteur à utiliser pour calculer une distance entre plans. On pourrait donc, éventuellement, concevoir une approche avec deux types descripteurs : la signature DCT pour la recherche et un autre descripteur plus robuste pour l'identification (une solution simple qui éviterait d'autres calculs serait d'utiliser l'information DCT non quantifiée). Toutefois, une première approche est d'utiliser la signature pour les deux tâches, car les calculs de distance entre signatures sont ultra-rapides et donnent des résultats satisfaisants (voir section 2.5).

La distance naturelle entre deux vecteurs binaires est la distance de Hamming. Cette distance mesure le nombre de bits différents entre deux vecteurs, c'est à dire que pour deux vecteurs binaires de même taille u et v la distance de Hamming est donnée par :

$$d_h(u,v) = \sum_i u[i] \oplus v[i] \quad \text{où } \oplus \text{ est l'opérateur binaire XOR}$$

Nous définissons, en premier lieu, une distance entre plans très simple, qui va nous servir à la fois à prouver la pertinence de la signature en tant que mesure de similarité, et nous servira ensuite de référence face à des méthodes plus évoluées lors des résultats.

Considérons deux plans $P_q = \{\sigma_{q_1}, ..., \sigma_{q_N}\}$ et $P_c = \{\sigma_{c_1}, ..., \sigma_{c_M}\}$, et supposons que le cas idéal $N = M$ a lieu. Une distance simple est la distance de Hamming moyenne entre signatures des plans P_q et P_c :

$$D_H(P_q, P_c) = \frac{1}{N} \sum_{i=1}^{N} d_h(\sigma_{q_i}, \sigma_{c_i})$$

La figure 2.7 montre l'efficacité de la distance de Hamming entre signatures. Elle est obtenue à partir d'une vidéo V de 24h dont on extrait un jingle J d'une longueur de 100 images. On construit alors le signal s tracé sur la figure par $s(i) = D_H(J, V_i)$ avec V_i la portion de vidéo de longueur 100 commençant à l'image i, $V_i = [I_i, I_{i+100}]$. On voit aisément trois instances de ce jingle dans le flux. La première instance est la requête elle-même (vers l'image 250.000) qui produit évidemment une distance nulle.

La distance de Hamming moyenne est donc une excellente mesure pour distinguer les répétitions, puisqu'elles sont aisément détectables par un simple seuillage. De plus, elle s'interprète facilement en tant que nombre d'erreurs binaire par signature. Par analogie avec le domaine des télécommunications, ce nombre d'erreurs binaire, normalisé par

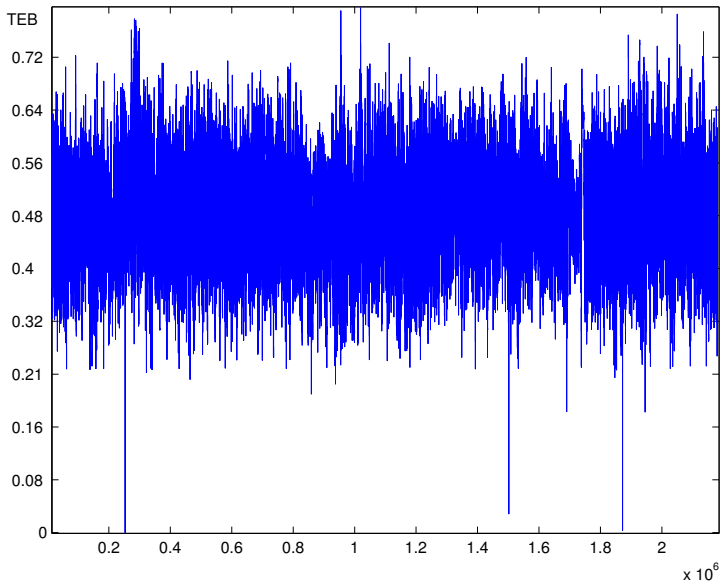

FIG. 2.7 – Distance entre un jingle et une vidéo de 24 heures

la taille de la signature, est appelé taux d'erreur bit (TEB). Le TEB a l'avantage de fournir une mesure de la distorsion indépendante de la taille de la signature et de la taille de la séquence. Le seuil TEB_{max} de l'algorithme 3 a alors une interprétation intuitive, et peut donc être fixé assez aisément.

Pour générer la figure 2.7, nous avons calculé l'ensemble des distances D_H sur le flux, autrement dit une recherche séquentielle. Ceci n'est pas envisageable, pour des raisons évidentes de complexité. L'intérêt majeur de l'algorithme 3 présenté dans la section précédente, est de réduire le problème de la recherche à un simple calcul de distance entre plans, et donc de se ramener à un problème beaucoup moins complexe.

La définition d'une telle distance pose malheureusement problème. Le plan requête et le plan candidat ont rarement le même découpage temporel. Ce mauvais découpage peut être causé par une erreur de la segmentation en plans, ou aux différents effets de transitions progressives qui modifient les images de début et de fin. Il s'agit alors d'aligner au mieux les deux plans afin que leurs signatures correspondent.

Les sections suivantes présentent des distances et des mécanismes qui permettent de résoudre ces problèmes.

2.4.2.2 Rapide état de l'art sur les méthodes d'alignement

Le problème de la définition d'une mesure de similarité temporellement robuste est un problème récurrent en vidéo. Une solution a été proposée par Adjeroh et al. [ALK99] qui définissent une mesure de similarité générique basée sur la distance d'édition. Ils modélisent les différents types de bruits auxquels leur distance doit être robuste par

différentes opérations d'éditions sensées être typique de la vidéo (insertion, suppression, substitution, inversion, fusion...). La distance finalement proposée est une combinaison de deux distances d'édition intégrant ces différentes opérations d'édition, l'une basée sur des attributs numériques, la seconde sur des attributs symboliques.

Une approche un peu semblable est proposée par Tan et al. [TKR99], qui définissent des contraintes d'alignement à respecter, et calculent le meilleur alignement en utilisant une technnique de programmation dynamique.

Des méthodes très semblables sont aussi proposées dans [LKE97] et [ZH06]. Elles utilisent aussi la distance d'édition afin de calculer le meilleur alignement suivant différentes contraintes et différentes métriques.

Des approches plus simples sont parfois proposées dans des cas où une telle robustesse n'est pas nécessaire. La détection de répétitions ne nécessite, en effet, pas forcément une distance capable de résister à des distorsions importantes (changement du nombre d'images par secondes, ré-ordonnancement d'images, etc.). En ce qui concerne les inter-programmes à la télévision, il ne semble pas que les diffusions successives d'un même inter-programme soit sujets à de fortes distorsions. Le problème est, par exemple, différent en radio, où des coupures, des accélérations, des superpositions peuvent se produire et compliquent singulièrement la tâche [Pin04].

Dans le contexte de la détection de répétitions vidéos, Pua et al. [PGGM04] utilisent une recherche exaustive afin de tester l'ensemble des alignements possibles entre deux plans. Une méthode encore plus simple est fournie par Shivadas et al. [GS06b], qui obtiennent directement un alignement grâce à une méthode de hachage perceptuel. Celle-ci met en correspondance une image du plan requête avec une image du plan candidat, et fournit donc implicitement une position pour l'alignement.

Nous proposons maintenant diverses méthodes qui permettent de résoudre ces problèmes.

2.4.2.3 Distance de Hamming non-alignée

Afin de pouvoir utiliser la distance de Hamming dans le cas où les plans sont de longueurs différentes, on peut simplement envisager de réduire le plan le plus long, afin qu'il ait la même longueur que le plus court.

Supposons avoir deux plans p et q, de longueur respectives N et M, où $N > M$. Cette réduction s'effectue, tout d'abord, en sous-échantillonnant le plan p par $\lfloor \frac{N}{M} \rfloor$. Cette valeur peut être égale à 1, auquel cas il n'y a pas d'échantillonnage. Nous obtenons alors un plan p' de longueur N'. La distance proposée ici est alors définie comme la distance de Hamming moyenne entre p' et q, calculée sur les seuls $min(N', M)$ premiers éléments. Les plans sont alors alignés à partir de leur première image. Nous appelons cette distance *non-alignée*, puisque l'alignement est, en fait, arbitraire.

Cette distance est définie essentiellement à des fins de comparaisons avec les autres distances définies plus loin. Cette méthode n'essayant pas de procéder à un quelconque alignement, ses performances sont attendues comme étant les moins bonnes.

2.4.2.4 Distance d'édition normalisée

La section 2.4.2.2 a montré que de nombreuses méthodes basées sur la distance d'édition [Lev65] ont déjà été proposées. Cette dernière semble être un bon candidat car elle cherche un alignement global entre deux séquences de descripteurs. La distance d'édition permet de comparer des séquences de longueurs différentes tout en minimisant de façon globale la somme des distances locales entre descripteurs. Cette distance locale est ici la distance de Hamming.

La distance d'édition peut, a priori, s'avérer utile dans notre cas, puisqu'il y a des suppressions et modifications d'images aux frontières des plans. Ces problèmes dûs aux transitions progressives créent des décalages qui sont bien gérés par une distance d'édition. Par contre, la distance d'édition n'est pas robuste à une erreur de segmentation en plan, car le problème devient alors une recherche de sous-séquence, qui se traite alors par des algorithmes spécifiques.

Un second problème est que les scores donnés par les distances d'édition ne sont pas comparables entre eux, car ils dépendent de la longueur des objets comparés. Prenons l'exemple de deux paires de séquences (X_1, Y_1) et (X_2, Y_2) qui diffèrent toutes deux en 2 positions, mais dans un cas les séquences sont de longueur $|X_1| = |Y_1| = 4$ et dans l'autre de longueur $|X_2| = |Y_2| = 100$. Ces paires ont la même distance d'édition alors que la première paire est dissimilaire et la deuxième très similaire. Ceci nous pose problème parce qu'il est alors difficile de seuiller la distance par une valeur de seuil fixe. Il existe toutefois des techniques appelées de post-normalisation qui consistent simplement à normaliser par la somme des longueurs des séquences [SC78]. Marzal et Vidal [MV93] ont malheureusement montré qu'une telle post-normalisation était sous-optimale et ont défini la distance d'édition normalisée (NED) qui permet d'avoir une distance cohérente quelle que soit la longueur des séquences.

Soit X et Y deux séquences de symboles, la NED est définie par :

$$NED(X,Y) = min\{\frac{W(P)}{L(P)} | \text{ P est un chemin de X à Y}\}$$

où $W(P)$ est la somme des poids le long du chemin P et $L(P)$ sa longueur. Les auteurs notent que la NED ne peut être calculée comme une post-normalisation de la distance d'édition, et que les techniques de post-normalisation[9] sont sous-optimales. La NED a montré qu'elle produisait des résultats supérieurs aux distances d'édition post normalisées.

La technique consiste à ajouter une dimension au calcul de la distance d'édition, en prenant en compte la longueur du chemin dans la minimisation. Bien qu'une technique de programmation dynamique similaire à celle de la distance d'édition soit possible [MV93], la NED reste d'une complexité élevée, de l'ordre de $O(|X| \cdot |Y|^2)$ (avec $|Y| \leq |X|$) ainsi qu'une complexité mémoire en $O(|X| \cdot |Y| \cdot (|X| + |Y|))$.

[9] utilisées par exemple en traitement de la parole

2.4.2.5 Recherche exhaustive

La méthode de recherche exhaustive est simple : elle teste tous les alignements possibles entre deux plans.

Soit un plan requête P_q et un plan candidat P_c de longueur respective N et M, $N < M$. La recherche exhaustive consiste à trouver la position qui minimise la distance de Hamming moyenne entre P_q et un sous-ensemble de P_c de longueur N. Cette position k_{min} est donnée par :

$$k_{min} = \text{Arg}\min_k \sum_{i=1}^{N} d_h(\sigma_{q_i}, \sigma_{c_{i+k}}) \quad \text{with } 0 \leq k \leq M - N$$

Cette recherche exhaustive n'autorise pas les débordements : le plan requête P_q est toujours inclu dans le plan candidat P_c. Cette méthode est appelée *recherche exhaustive sans débordements*.

Nous définissons une variante de cette méthode, qui considère un nombre d'alignements plus élevés, en autorisant que le plan requête « déborde » du plan candidat. Ceci permet d'aligner seulement une partie des plans, ce qui peut être utile en cas de mauvaise segmentation en plans, ou de transitions progressives. Il faut, toutefois, imposer une borne sur la longueur du débordement maximal autorisé, sans quoi l'alignement n'aurait plus vraiment de sens. Cette borne impose donc que la partie commune entre les deux plans ne soit pas trop faible. La longueur du débordement maximal est choisi égal à $N/2$. La figure 2.8 illustre les positions extrêmes d'alignement entre le plan candidat et le plan requête. Cette méthode est appelée *recherche exhaustive avec débordements*.

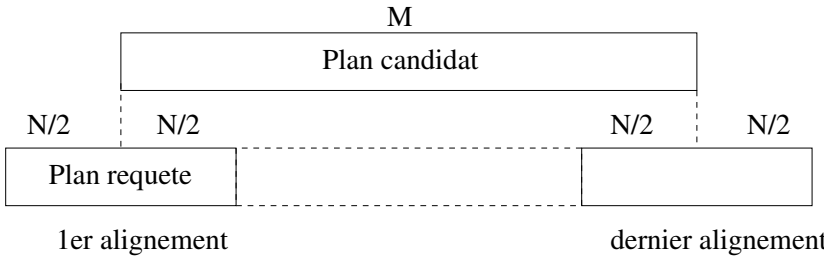

FIG. 2.8 – Alignements considérés par la recherche exhaustive avec débordements

Bien qu'elle puisse paraitre complexe, la recherche exhaustive peut être assez légère, en particulier si le premier plan candidat est correct. Dans ce cas, on peut rapidement trouver un alignement correct, c'est à dire un alignement qui génère une distance inférieure au seuil TEB_{max}, valeur pour laquelle on considère que les plans sont des répétitions. La recherche est alors stoppée dès que la distance tombe en-dessous de ce seuil.

En revanche, si la méthode de recherche propose de nombreux plans qui se révèlent erronés, la recherche exhaustive peut alors être complexe, en particulier si les plans candidats ou requêtes sont de grande taille, car de nombreux alignement différents sont alors testés. La complexité de la recherche exhaustive dépend donc du taux de fausses alarmes de la signature, et donc, indirectement, de la composition de l'EVR. La complexité de la recherche exhaustive est donc difficilement prévisible.

2.4.2.6 Distance ancrée

Cette distance est basée sur une constatation simple : le plan candidat est obtenu par le biais d'une signature σ_{c_i} qui a été reconnue identique à une signature σ_{q_j} du plan requête, c'est à dire $\sigma_{c_i} = \sigma_{q_j}$. Si les positions relatives des signatures dans leur plan respectif, i et j, ont été conservées, alors nous pouvons aligner les plans en prenant la position de reconnaissance comme une *ancre*. Nous appelons donc cette distance la *distance ancrée*. La figure 2.9 donne un exemple d'un tel alignement, où est mise en évidence la partie commune aux deux plans. La distance ancrée est définie comme la distance de Hamming moyenne sur cette partie commune.

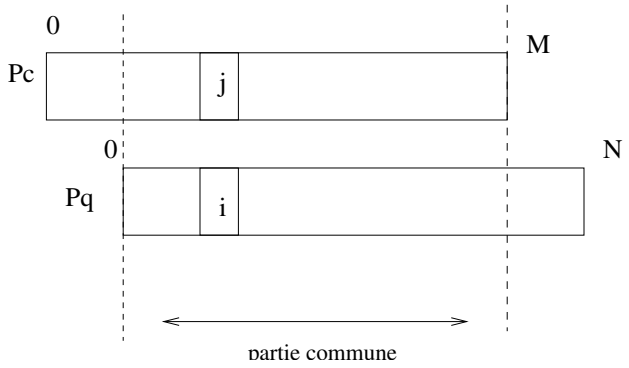

FIG. 2.9 – Alignement avec la distance ancrée

Il est important de remarquer que plusieurs alignements peuvent être testés au cours du processus global de détection. L'algorithme 3 stipule que toutes les signatures sont testées tant que l'une d'elles ne renvoie pas un couple (plan, position) qui soit en dessus du seuil. Plusieurs, ou même la totalité des alignements peuvent ainsi être testés au cours de la détection.

Une autre remarque importante est qu'il n'y a aucune condition d'inclusion d'un plan dans l'autre, l'alignement étant seulement basé sur la position de la signature reconnue, sans se soucier du fait que les plans aient une partie commune importante. Cette distance accepte donc les débordements.

A noter que cette même méthode a été utilisée par [GS06b] pour la reconnaissance de publicité, en utilisant de la même manière que nous une information de position provenant d'une méthode de hachage perceptuel.

2.5 Résultats

Cette section présente les résultats de la méthode de détection des répétitions. Il est difficile d'évaluer les différents paramètres séparément, puisqu'ils contribuent tous aux performances de l'algorithme global. La signature et ses différents paramètres ont, en particulier, une grande influence sur les résultats. Nous nous concentrons dans la section 2.5.1 sur la signature, afin d'effectuer un choix, mais ce choix ne peut être définitif sans

une évaluation des méthode d'organisation de la table de hachage, évaluées en section 2.5.2, où la signature a une influence non négligeable sur les résultats.

Une fois le choix de la signature et la méthode d'organisation de la table de hachage effectué, les différentes méthodes d'alignement des plans sont évaluées en section 2.5.3. Enfin, la section 2.5.6 évalue la complexité et teste le passage à l'échelle.

2.5.1 Choix de la signature

Nous cherchons, en premier lieu, à vérifier la propriété d'invariance de la signature. Nous rappelons que, dans l'algorithme 3, nous avons fait une hypothèse assez forte, qui est que les segments répétés partagent au moins une signature **identique**. La véracité de cette hypothèse est testée dans la section 2.5.1.1 sur des données réelles. Un choix est ensuite effectué sur la taille de la signature en section 2.5.1.2, et sur le schéma de quantification en section 2.5.1.3.

2.5.1.1 Test de l'invariance

Nous testons dans cette section l'invariance de la signature dans une situation réelle et, en particulier, le pourcentage de signatures effectivement invariantes. Nous souhaitons aussi estimer le nombre de fausses alarmes que génère une indexation directe.

Nous définissons le protocole expérimental de cette expérience, que nous appelons *expérience0*. C'est le corpus 1 qui est utilisé[10], il est composé d'une vidéo d'une heure, *video_1h*, et d'une vidéo de 24 heures, *video_24h*, enregistrées respectivement le 16/11/2004 et le 15/11/2004. Leur proximité temporelle fait qu'il existe de nombreuses répétitions entre ces deux vidéos. L'expérience consiste à chercher chaque signature de la video *video_1h* dans la table de hachage, qui contient les signatures de *video_24h*. Si la signature ne renvoie pas de plan candidat alors qu'il existe une répétition, ceci est alors compté comme un manqué. Si la signature renvoie un (ou plusieurs dans le cas de la référence multiple) plan(s) candidat(s), il est ensuite vérifié si au moins un des plans candidats obtenu est correct ou non. Les résultats sont donnés sous forme de rappel et précision dans la table 2.6 pour les deux types de références : simple et multiple.

	Référence simple		Référence multiple	
n	Précision	Rappel	Précision	Rappel
8	98.7	79.3	96.5	98.4
7	98.3	85.2	97	99.2
6	94.1	90.3	95.7	99.6
5	58.9	94.1	70	99.9

TAB. 2.6 – Test d'invariance de la signature dans un cas réel

Il est important de comprendre que cette table donne les taux de rappel et de précision *par signature*. La précision donne une indication du nombre de fausses alarmes au niveau signature, ce qui se traduit au niveau plan par une estimation du nombre

[10] voir la présentation du corpus en annexe A

d'essais avant de trouver un plan correct. De même, le rappel mesure le pourcentage de signatures qui ne génèrent pas de plan candidat. En pratique, il suffit d'une seule signature par plan qui renvoie un plan correct pour que la méthode fonctionne, mais plus la précision est élevée, et plus le nombre de distances à calculer entre plan requête et plans candidats sera faible. De même, plus le rappel est élevé, et plus le nombre de requête sans résultats sera faible. C'est donc la précision qu'il est important de maximiser pour réduire la complexité, car on réduit alors le nombre de distances entre plans à calculer. Le rappel est moins critique et peut avoir une valeur assez peu élevée sans conséquences importantes.

Une étude du tableau 2.6 montre, en conséquence, que les signatures de plus grande taille sont préférables. Elles obtiennent une meilleure précision, tout en ayant un rappel correct. La comparaison entre méthodes par référence multiple et référence simple tourne clairement en faveur de la référence multiple : le taux de rappel est bien plus élevé, ce qui était un résultat attendu, tout en ayant une précision légèrement plus faible. Il est en revanche trop tôt pour conclure sur la supériorité de la méthode par référence multiple, car ces résultats sont au niveau signature, et rien n'indique qu'un très haut rappel au niveau signature soit indispensable au niveau plan.

2.5.1.2 Taille de la signature

Cette section a pour but de choisir la taille de la signature. Nous définissons, tout d'abord, deux expériences.

La première expérience, appelée *expérience1*, consiste à réaliser l'ensemble du processus de détection des répétitions, tel que présenté par l'algorithme 3. Cette expérience est réalisée sur le corpus 1, en utilisant *video_1h* comme requête et *video_24h* comme EVR. La vérité terrain indique que 147 plans de *video_1h* sont aussi présents dans *video_24h*.

L'*expérience2* est exactement la même qu'*expérience1*, les vidéos *video_24h* et *video_1h* intervertissent simplement leur rôle de requête et d'EVR. La vérité terrain indique que 497 plans de *video_24h* sont aussi présents dans *video_1h*.

Ces expériences comportent plusieurs paramètres à déterminer : la méthode de quantification, la distance entre plans, la taille de la signature, ainsi que la méthode d'organisation des données (référence simple ou multiple). Ici, nous fixons arbitrairement la méthode de quantification comme étant la méthode de quantification par la valeur médiane, et la distance entre plans est la recherche exhaustive sans débordements. L'expérience effectuée est l'*expérience1*.

La table 2.7 donne les résultats pour des tailles de signatures variant de 25 à 64 bits, et pour les méthodes d'indexation par référence simple et par référence multiple.

Les résultats du tableau 2.7 confirment ceux du tableau 2.6 en ce qui concerne l'influence de la taille de la signature sur les résultats. Une signature de 64 bits, c'est à dire $n = 8$, obtient une excellente précision, ainsi qu'un très bon rappel. Il était attendu qu'une signature de plus grande taille obtienne une meilleure precision puisqu'elle comporte plus d'information, sa capacité de discriminance est meilleure. Il n'était par contre pas évident qu'elle obtiennent un aussi bon rappel. Le choix de la valeur de n est un

n	Référence simple		Référence multiple	
	Précision	Rappel	Précision	Rappel
8	99.3	97.3	98.6	97.3
7	96.6	97.2	96.6	97.2
6	92.6	96.5	92.7	97.2
5	70.3	97.9	68.1	98.6

TAB. 2.7 – Résultats de la détection des répétitions en fonction de n et de la méthode de référence. Version recherche exhaustive sans débordements.

compromis à faire entre une grande valeur, qui augmente le pouvoir discriminant, mais réduit la possibilité d'une indexation directe car il y a alors moins de chances pour que les signatures soient identiques.

La figure 2.10 donne un autre aperçu du problème, en faisant varier n sur de plus grandes plages de valeurs. Ces courbes sont obtenues à partir de deux séquences d'images, répétitions l'une de l'autre, sur lesquelles sont extraites des signatures pour différentes de valeur n et sur lesquelles la distance de Hamming est calculée. La partie basse de la figure, qui donne les résultats en terme de TEB, est particulièrement intéressante. On peut y voir que les signatures de petite taille ($n < 8$) subissent un surcoût important en terme de TEB. Le compromis est alors de déterminer la valeur de n qui ne subit pas ce surcoût, tout en étant la plus petite possible pour permettre une indexation directe.

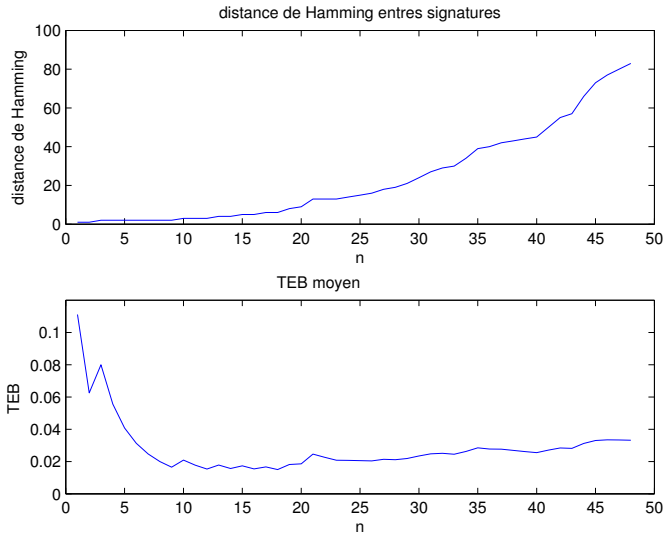

FIG. 2.10 – Distance de Hamming en fonction de la taille de la signature

L'ensemble de ces résultats nous conduit à choisir une valeur de $n = 8$, soit une signature de 64 bits pour la suite du travail. Un argument supplémentaire dans le choix

d'une signature de 64 bits est que la signature est alors simplement représentée par un entier 64 bits, ce qui simplifie grandement l'implémentation. En particulier, la fonction de hachage définie à la section 2.3.2.2 est implémentable à l'aide d'un simple `cast` d'un entier 64 bits vers un entier 32 bits.

2.5.1.3 Comparaison des schémas de quantification

Nous cherchons dans cette section à déterminer le meilleur schéma de quantification parmi les deux proposés en section 2.2.3. Pour cela, nous reprenons l'*expérience1* et l'*expérience2*, avec comme paramètres : une signature de taille 64 bits, la distance entre plans est la distance ancrée, et l'organisation de la table de hachage est par référence simple.

Le tableau 2.8 donne les résultats. $Q1$ fait référence au schéma qui binarise les signatures à partir de leur valeur médiane, tandis que $Q2$ code la signifiance. $Q1$ possède un rappel plus élevé que $Q2$ mais semble un peu moins robuste. La différence en terme de rappel semble toutefois indiquer que $Q1$ est meilleur, bien que la différence ne soit pas réellement significative. Il serait nécessaire de tester ces deux schémas de quantification avec une vérité terrain plus importante pour réellement les départager.

	Expérience1 (147 répétitions)		Expérience2 (497 répétitions)	
Méthode	Précision	Rappel	Précision	Rappel
Q1	99.3	98	100	96.6
$Q2$	100	96.6	100	95.1

TAB. 2.8 – Résultats des schémas de quantification

Le schéma de quantification $Q1$ est utilisé comme le schéma de quantification par défaut dans la suite des travaux.

2.5.2 Organisation des données

Concernant la méthode d'organisation des signatures dans la table de hachage, les résultats précédemment présentés dans le tableau 2.7 sont à l'encontre de ce que l'on attendait. On aurait pu croire, en effet, d'après les premiers résultats de la section 2.5.1.1, qui donnaient un net avantage à la méthode par référence multiple, que la méthode par référence multiple était globalement meilleure. Ce n'est pas le cas : le rappel est quai identique à celui de la méthode par référence simple, et la précision est inférieure. Sachant que les références multiples nécessitent considérablement plus de mémoire, et sont de plus, plus lentes, nous choisissons d'abandonner le système de référence multiple au profit de la référence simple.

Le tableau 2.9 ne fait que confirmer ces résultats dans le cadre d'un alignement des plans par la méthode de recherche ancrée. L'explication de la supériorité de la méthode par référence simple au niveau du plan est, en fait, assez simple. Il faut toutefois se rappeler que dans le cas des résultats de l'*expérience0*, la précision et le rappel étaient

n	Référence simple		Référence multiple	
	Précision	Rappel	Précision	Rappel
8	99.3	98	98.6	98
7	95.9	98.6	95.9	98.6
6	92.2	98.6	92.2	98.6
5	66.3	97.9	62.9	97.2

TAB. 2.9 – Résultats de la détection des répétitions en fonction de n et de la méthode de référence. Version recherche ancrée

calculés pour chaque signature. Dans l'expérience1, ces mesures sont définies au niveau du plan. La précision au niveau de la signature était déjà supérieure dans le cas référence simple dans l'*expérience0*. Les résultats ne sont donc pas surprenants en ce qui concerne la précision. L'amélioration du rappel s'explique par le fait qu'individuellement, les signatures obtiennent un taux de rappel assez faible, mais à l'échelle du plan, les signatures proposent suffisament d'hypothèses (y compris la bonne, d'après les résultats) pour pouvoir se passer d'une méthode par référence multiple.

2.5.3 Comparaison des méthodes d'alignement des plans

Cette section étudie les différentes distances entre plans définies en 2.4.2. Les expériences 1 et 2 sont réalisées avec les paramètres par défaut (signature 64 bits, référence simple, quantification par la valeur médiane). Cinq distances sont testées : les deux distances par recherche exhaustive, la distance ancrée, la distance non-alignée, et la NED. Notons que ces distances sont aisément comparables, car elles fournissent toutes une distance interprétable en terme de TEB.

Les résultats sont présentés dans la figure 2.11, pour les expériences 1 et 2, où le paramètre variable est le seuil de la distance entre plans, appelé TEB_{Max} dans l'algorithme 3. Les résultats sont obtenus en faisant varier TEB_{Max} de 0.047 (3/64) à 0.4375 (28/64), par pas de 1/64.

La hiérarchie attendue est globalement respectée. On peut toutefois être surpris et déçu par les performances de la NED. Celle-ci obtient les plus mauvais résultats sur l'expérience1, et des résultats inférieurs aux deux distances exhaustives et à la distance ancrée sur l'expérience2, avec toutefois un assez bon rappel. Une étude détaillée des fausses alarmes de la NED montre en fait que ses « erreurs » sont sujettes à discussion, car leur statut d'erreur dépend du sens que l'on donne à la notion de répétition. Dans notre définition des répétitions, nous autorisions quelques éditions mineures. La NED est capable de trouver des éditions que nous considérons comme trop importantes pour que les séquences soient des répétitions selon notre définition. De la même manière, la NED peut trouver des séquences qui ne sont pas des transformations l'une de l'autre, mais qui sont très similaires, l'exemple typique étant les plans de présentateurs d'émissions récurrentes (journal, jeux).

En ce qui concerne les autres distances, les résultats sont moins suprenants. La distance non-alignée obtient des résultats qui restent plutôt bons, mais inférieurs aux

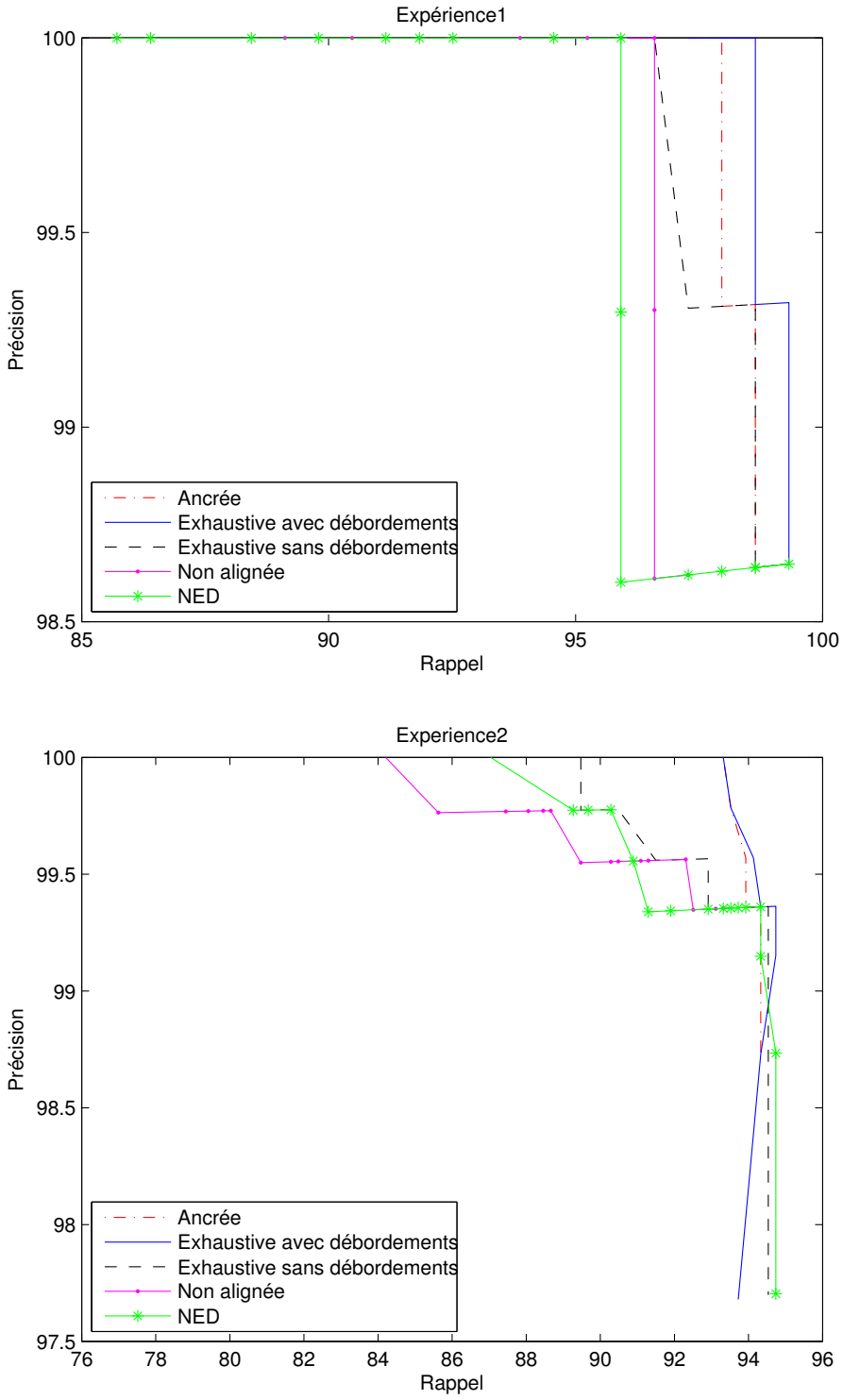

FIG. 2.11 – Courbes de précision/rappel pour les différentes méthodes d'alignement. La figure supérieure concerne les résultats de l'expérience1, la figure inférieure ceux de l'expérience2.

autres distances. La recherche exhaustive avec débordements est clairement la meilleure, en particulier, par rapport à sa version sans débordements. Ceci montre que de nombreux problèmes se situent aux bornes des plans, et que la possibilité de réaliser un alignement partiel (en ne considérant qu'un sous-ensemble d'un plan) permet clairement d'améliorer les résultats. La distance ancrée, qui elle aussi autorise les alignements partiels, se rapproche des résultats de la recherche exhaustive avec débordements, ce qui tend à montrer que la position de la signature permet un alignement correct, proche de l'optimal.

Nous étudions maintenant ces distances d'un point de vue complexité. La figure 2.12 donne les temps de recherche pour chacune des cinq distances. Le protocole expérimental consiste à rechercher l'ensemble des plans d'une vidéo de 24 heures dans un EVR de taille croissante, de 24 à 500 heures. Nous mesurons le temps pour effectuer la totalité du processus de recherche, défini par l'algorithme 3. La figure montre que les temps de recherche de la NED et des deux types de recherche exhaustive sont bien plus élevés que la méthode ancrée et la distance non-alignée, qui ont des temps de recherche de l'ordre de la demi-seconde, quelque soit la taille de l'EVR.

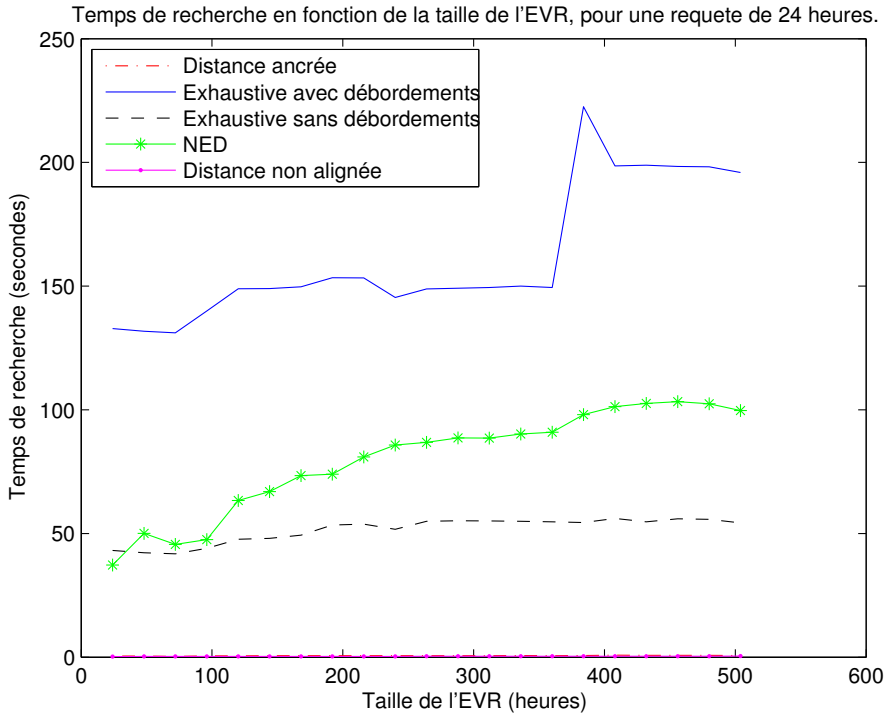

FIG. 2.12 – Temps de calcul pour les différentes méthodes d'alignement

Pour conclure, parmi ces cinq distances, deux retiennent notre attention. La distance exhaustive avec débordements, qui obtient les meilleurs résultats en terme de précision et rappel, mais qui est d'une complexité élevée. La deuxième distance remarquable

est la distance ancrée, qui obtient des résultats légèrements inférieurs à la distance exhautive avec débordements, mais avec une complexité bien plus faible. La qualité des résultats de cette distance est, à notre avis, dûe à sa capacité à gérer les débordements, ainsi que la robustesse de l'information de position de la signature. Ces résultats sont confirmés par Shivadas et al. [GS06b] qui, en alignant aussi les plans grâce à la position de sa signature, obtient de meilleurs résultats qu'une recherche exhaustive (a priori sans débordements...). En terme de complexité, il semble en tout cas difficile de faire plus simple, et donc plus rapide.

Au vu de l'ensemble des résultats, nous choisissons d'utiliser la distance ancrée comme distance par défaut dans la suite du travail, en raison de sa faible complexité. Toutefois, la recherche exhaustive avec débordement peut aussi être un bon choix, lorsque la complexité n'est pas un problème. La NED peut aussi présenter un intérêt, dans le cas où une plus grande robustesse est souhaitée.

2.5.4 Remarque sur la redondance de l'EVR

Il est important de remarquer que le rappel est meilleur lorsque l'EVR est video_24h. Ce résultat s'explique par la redondance présente dans video_24h. Cette dernière est une journée de télévision enregistrée en continu, sans aucune modification, et contient donc de nombreuses répétitions. La redondance de l'EVR est bénéfique, car l'EVR contient des répétitions au sens où nous l'avons défini : ce ne sont pas des copies identiques mais des copies modifiées par la transmission. Ces répétitions génèrent des signatures différentes, et permettent donc d'augmentent le rappel.

2.5.5 Analyse qualitative des résultats

FIG. 2.13 – Exemple de déformations auxquelles la signature n'est pas robuste. TEB de 0.2 entre les 2 images.

Cette section analyse de manière qualitative les résultats de détection, et donne

quelques explications sur les fausses alarmes et leur criticité. Nous déterminons aussi la valeur du seuil de détection.

Le seuil de détection, appelé TEB_{max}, est celui qui permet de prendre la décision finale quant au fait que le plan candidat est effectivement une répétition, en fixant la distance maximale admissible entre plans. Nous avons déjà mentionné que cette distance, lorsqu'elle est normalisée par rapport à la longueur du plan, possède une interprétation simple : c'est le taux d'erreur binaire (TEB), c'est à dire le nombre d'erreurs par bit. Nous pouvons alors nous reporter aux figure 2.7, 2.10, et la table 2.4, afin d'avoir l'intuition sur le rapport entre le TEB et la déformation. Nous avons déjà signalé, en section 2.2.4, que pour deux images non corrélées la valeur moyenne du TEB est de 0.5, soit une distance de Hamming moyenne de 32, puisque la signature est de 64 bits. Un plan ayant un TEB moyen proche, ou au-dessus, de 0.5 peut donc être considéré comme n'étant pas une répétition du plan original. A titre d'information, le TEB varie entre 0 et 0.16 pour des images successives d'un même plan. Certaines déformations produisent des distances de Hamming assez élevées, comme celle présentée en figure 2.13, où l'ajout de deux bandeaux d'assez grande taille, en plus d'une variation colorimétrique génère un TEB de 0.2.

La signature génère aussi des fausses alarmes, en particulier sur les images de syntèse comportant peu d'information. La DCT n'est en effet pas adaptée pour représenter ce genre d'images, où l'énergie est alors bien plus distribuée entre les coefficients et non plus concentrée dans les basses fréquences. La figure 2.14 donne plusieurs exemples d'images naturelles et de synthèse, ainsi que leur transformée DCT. On voit particulièrement bien sur cette dernière figure que, pour les images comportant peu d'information, l'énergie est dispersée (*RTL*), ou alors l'énergie est très faible (*france télévision*). La signature n'est alors plus caractéristique de l'image et peut être proche en distance de Hamming d'une signature d'une image différente, comportant elle aussi peu d'information.

Les images *france télévision* et *RTL* données en exemple ont toutefois des logos d'assez grande taille. Même si cette information est relativement mal représentée par la DCT et donc par la signature, les distances de Hamming entre deux répétitions d'une de ces séquences seront en général assez faibles. Par contre la propriété d'invariance est plus difficile à respecter, et un simple bruit peut alors inhiber la propriété d'invariance. Ce genre de séquence est donc relativement souvent manqué. Il en est de même pour les plans comportant des mouvements très rapides sur une grande partie de l'image : la signature ne permet pas de coder ce genre d'image de façon fiable. Ce genre de plans peut donc aussi être manqué.

L'ensemble de ces considérations nous conduit à choisir une valeur de seuil $TEB_{max} = 0.0625$. Malgré ce seuil assez bas pour éviter les fausses alarmes, certaines subistent toujours, sur des images presque totalement monochromes, avec pas ou très peu d'éléments distinctifs. Il s'agit typiquement des images de transitions entre scènes d'un téléfilm ou des images de séparation entre inter-programmes. Il peut donc être utile de filtrer les plans totalement monochromes, afin d'éviter des détections parasites. En pratique, toutefois, les fausses alarmes sont très rares.

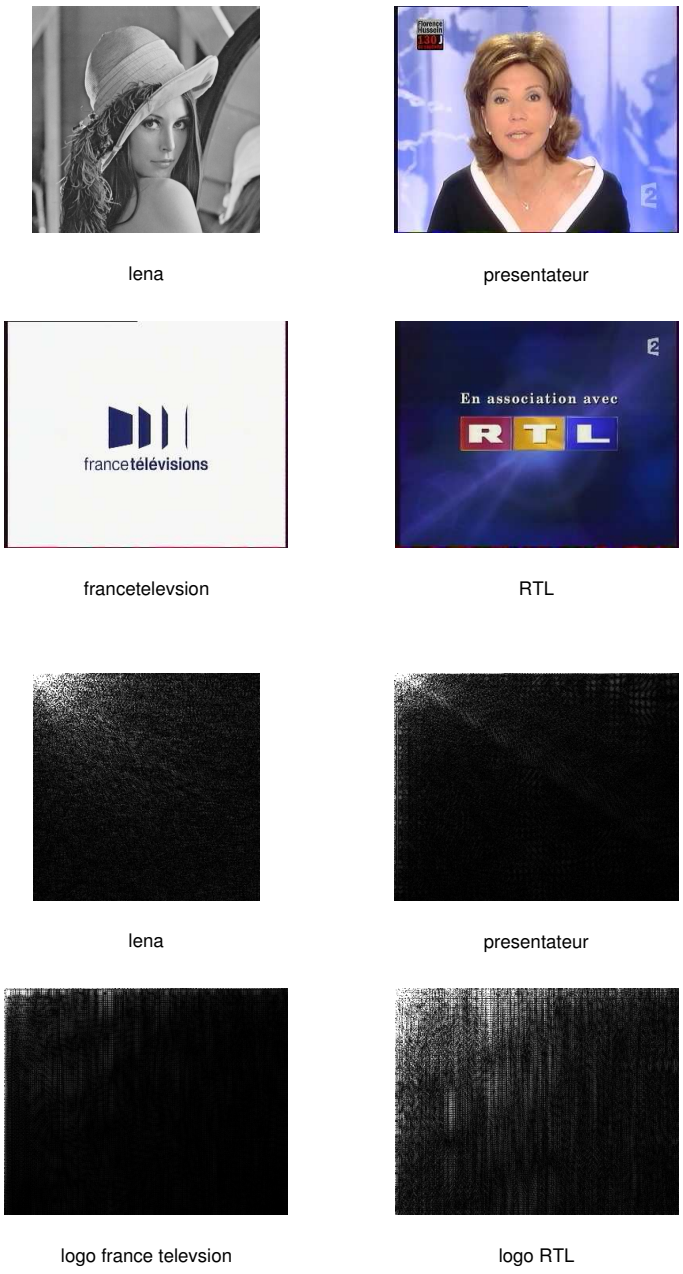

FIG. 2.14 – Exemple d'images naturelles et synthétiques, et leur transformée DCT.

2.5.6 Complexité et passage à l'échelle

La rapidité de la recherche de répétitions est un facteur clé de l'algorithme. Cette section évalue expérimentalement la complexité de la méthode proposée sur des en-

sembles de données assez importants, afin de vérifier que la méthode est effectivement capable d'obtenir une vitesse de recherche constante, quelle que soit la taille de l'EVR considéré.

Nous utilisons les paramètres choisis dans les sections précédentes comme étant les plus appropriés. Pour rappel, nous utilisons une signature de 64 bits quantifiée par la méthode de la valeur médiane, les signatures sont stockées dans la table en utilisant une simple référence, la distance entre plans est la distance ancrée et la valeur du seuil utilisée pour seuiller cette distance est $\alpha = 4$.

La détection des répétitions se fait en deux phases : extraction des caractéristiques et recherche des répétitions. L'extraction des caractéristiques comporte le décodage vidéo, la segmentation en plans et le calcul des signatures, ce qui produit, pour une journée de 24 heures, un fichier de métadonnées de l'ordre de 17 Mo, soit 0.04% de la taille de la vidéo originale. la détection des répétitions se fait ensuite entre deux fichiers de métadonnées. Les processus de calcul de caractéristiques et de recherche sont donc bien séparés. Les deux processus sont distingués pour l'évaluation de performances.

Les expériences sont réalisées sur un PC mono-processeur 3Ghz et 1Go de Ram, sous Linux Fedora 4. La librairie FFTW [FJ05] est utilisée pour calculer la DCT. Le corpus utilisé est le **corpus2**, présenté en annexe A.

L'extraction des caractéristiques est bien plus rapide que le temps réel puisque nous arrivons à des vitesses de traitement de l'ordre de 115 images/s, décodage inclu. Pour information le décodage seul s'effectue à une vitesse de 270 images/s. Le traitement d'une vidéo de 24 heures ne dure alors qu'un peu plus de 5 heures. On peut aussi se rapporter au tableau du début du chapitre, section 2.2.2.2, pour un détail de la complexité en temps des deux phases d'extraction de caractéristiques[11].

Les sections suivantes étudient la complexité de la phase de recherche des répétitions, qui, nous l'avons dit, nécessite uniquement les fichiers de métadonnées générés par l'extraction des caractéristiques, et n'a donc pas besoin d'accéder au flux vidéo.

2.5.6.1 Étude de la distance ancrée

Dans cette section, nous étudions le comportement de la distance ancrée en terme de complexité. La première expérience effectuée consiste à détecter les répétitions entre chaque paire de fichiers du corpus2, c'est à dire que chacune des 21 journées est recherchée dans toutes les autres journées. Nous mesurons alors deux paramètres, le nombre de détection et le temps de calcul (temps de recherche). Les résultats sont donnés sur la figure 2.15.

Ces résultats ne sont pas faciles à interpréter puisqu'ils ne sont pas linéaires, des points aberrants apparaissent sur le côté droit de la figure. Certains tests prennent en effet jusqu'à 9 secondes, sans que cela soit relié au nombre de détections. Le seul élément qui nécessite un calcul dans l'algorithme 3 est l'alignement, autrement dit, la distance entre plans. C'est donc le nombre de distances calculées qui fait augmenter le temps de recherche. Dans le meilleur des cas, chaque calcul de distance résulte en une détection, mais en pratique, il existe des fausses alarmes au niveau signature. De nombreuses

[11] En sachant que la case *(avec filtrage)* comprend, en fait, l'ensemble des traitements.

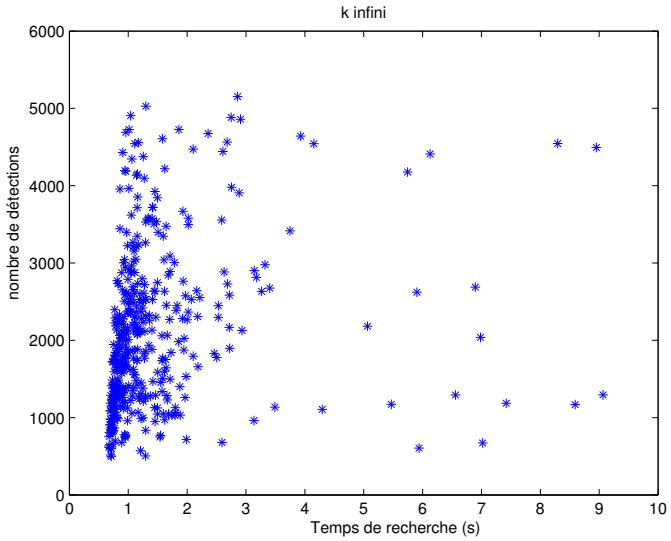

FIG. 2.15 – Nombre de détections en fonction du temps de calcul, $k = \infty$

distances sont donc calculées sans pour autant générer de détections. Le nombre de distances calculées dépend donc de la robustesse de la signature. Nous rappelons que ces fausses alarmes signatures peuvent être de deux types :

- la signature renvoie un plan candidat erroné
- la signature renvoie un plan candidat correct mais avec une position erronée

Il peut donc y avoir de nombreux essais avant que la bonne position ne soit trouvée. Un problème critique est lorsque deux plans partagent un grand nombre de signatures identiques. Des distances entre ces plans sont donc calculées, mais les différents alignements ne font jamais descendre la distance en dessous du seuil de détection. C'est ce type de situation qui fait augmenter le calcul de manière importante, et introduit de la variations dans les temps de recherche, à nombre de détection constant. Nous appelons ce phénomène de la *gigue*. Pour résoudre ce problème, nous définissons un nombre d'essais maximal k pour un plan candidat. La figure 2.16 montre alors l'évolution du temps de recherche avec k. Le cas particulier où le nombre d'essais n'est pas contraint est le cas $k = \infty$, et est donné par la figure précédente, figure 2.15.

La figure 2.17 donne le lien entre nombre de répétitions et le temps de recherche, pour différentes valeurs de k. Pour $k = 1$, nous voyons clairement apparaitre une dépendance linéaire entre le nombre de détection et le temps de recherche, ce qui confirme que ce sont bien les essais infructueux qui génèrent de la gigue. En pratique, $k = \infty$ peut toutefois être une valeur acceptable, car le temps de recherche est toujours inférieur à 10 secondes, ce qui, vu les masses de données considérées, peut être tout à fait acceptable. En revanche, si une réponse rapide est souhaitée et/ou si la gigue n'est pas acceptable, une valeur de $k = 20$ semble un choix raisonnable, d'après la figure 2.16, car elle permet de limiter ces deux facteurs sans perdre en nombre de détections.

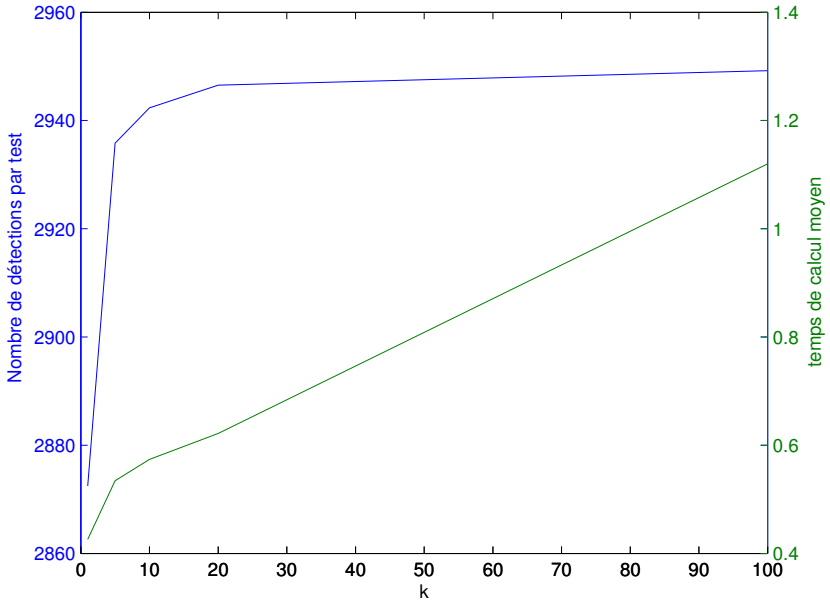

FIG. 2.16 – Évolution du nombre moyen de détection et du temps de calcul moyen, en fonction de k

2.5.6.2 Passage à l'échelle

La nature de l'algorithme de recherche implique, en théorie, que le temps de recherche est constant, quelque soit la taille de l'EVR. Nous essayons dans cette partie de vérifier cette affirmation, en vérifiant que la méthode permet bien de gérer un grand ensemble de vidéos. Notons dès maintenant que, dans notre application, l'EVR contiendra l'ensemble des inter-programmes et des génériques de programmes, qui pourront aider la structuration. Cet ensemble n'a pas besoin d'être gigantesque. Au contraire, les inter-programmes et les génériques étant courts, il est probable que l'EVR n'atteindra pas des tailles démesurées. Il est probable qu'il soit toutefois d'une taille conséquente, et nous estimons que travailler avec des EVR de 200 à 500 heures est réaliste dans un contexte applicatif.

La figure 2.18 montre les résultats de la détection des répétitions entre une requête de 24 heures et un EVR variant entre 24 et 504 heures. Les résultats sont donnés à la fois en nombre de détections, autrement dit, le nombre de plans répétés entre la requête et l'EVR, et en temps. La figure de gauche est obtenue avec une requête enregistrée le 28 mars 2005, soit une semaine plus vieille que l'EVR. La figure de droite est obtenue avec **video_24h** conmme requête, qui est 6 mois plus ancienne que l'EVR. La distance ancrée est utilisée avec un facteur limitant le nombre d'essais, $k = 20$.

Deux remarques viennent à l'esprit lorsque l'on contemple ces figures : le temps de recherche n'est pas constant quand la taille de l'EVR augmente, et les figures de droite et gauche sont très différentes, bien que ce soit la même expérience. Ces deux remarques

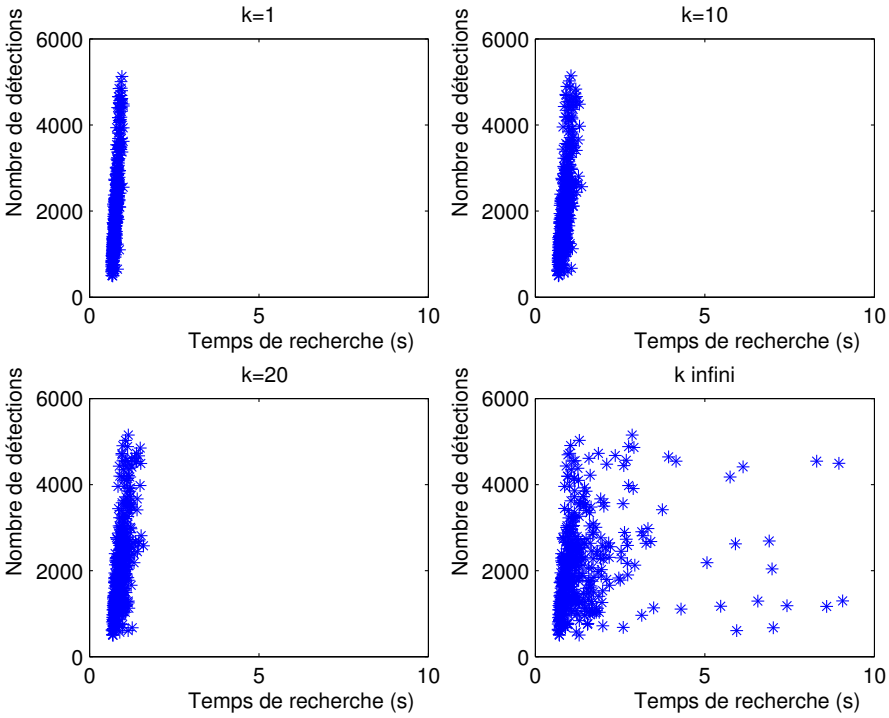

FIG. 2.17 – Influence du paramètre k sur le temps de recherche

peuvent en fait être expliquées assez simplement, par les remarques faites dans la précédente section. C'est le nombre de distances calculées qui fait augmenter le temps de recherche. Le nombre de distances calculées est aussi le nombre de signatures requêtes qui renvoient un plan candidat. L'augmentation de la taille de l'EVR augmentant la probabilité de fausses alarmes signatures, il est donc normal que le temps de recherche augmente en moyenne avec la taille de la base.

Toutefois, c'est aussi le contenu qui fait varier les résultats de recherche. La différence entre la figure de gauche et celle de droite est importante, tant en terme de nombre de détections que de temps de recherche, alors que la seule différence est dans le contenu de la requête. C'est aussi le contenu qui explique la non-linéarité des courbes de temps.

En ce qui concerne la complexité, les temps de recherche de répétitions entre une requête de 24 heures et un EVR de 24 à 500 heures restent toujours aux alentours de 1 seconde, ce qui est extrêmement rapide, sachant qu'une requête de 24 heures équivaut environ à 20 000 sous-requêtes. Il est difficile de comparer avec d'autres méthodes, du fait des différences de but, de données, et des tailles de base considérées, mais notre méthode est clairement parmi les plus rapides.

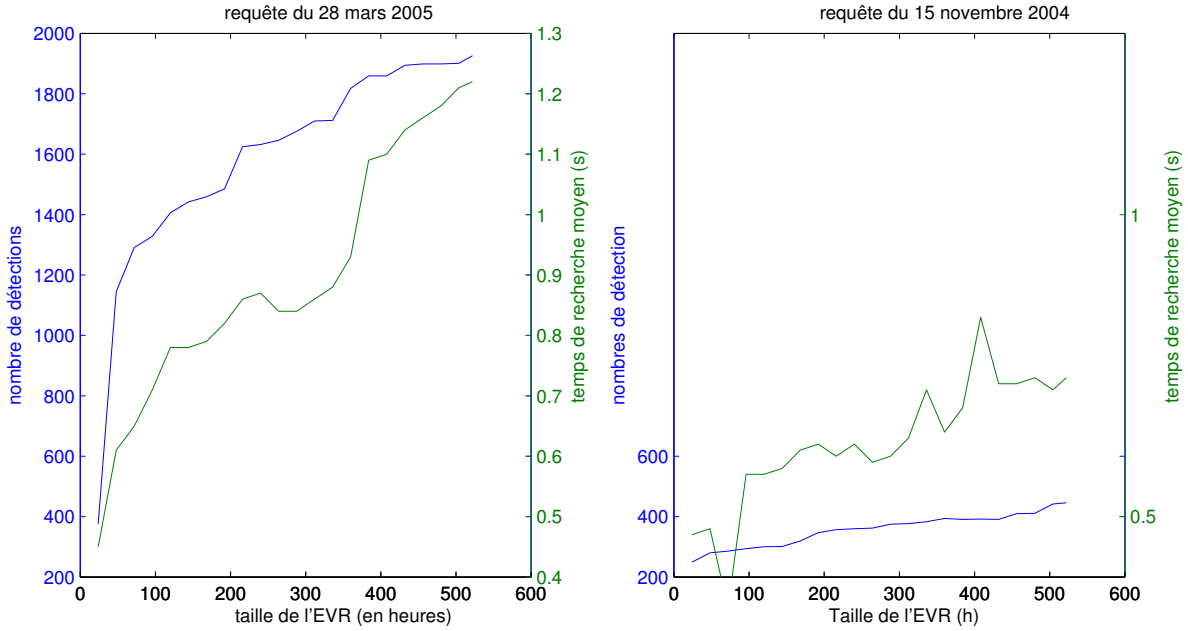

FIG. 2.18 – Influence de la taille de l'EVR et du nombre de détections sur le temps de recherche

2.6 Synthèse

Nous avons proposé dans ce chapitre une méthode de détection des répétitions entre deux vidéos de grande taille, en considérant conjointement le problème de la description et de la recherche. La philosophie de la méthode est d'utiliser une *indexation directe*, qui permet d'obtenir très rapidement des candidats, en un temps constant. Cette indexation directe est possible grâce à une signature compacte mais, néanmoins, discriminante, construite à partir de coefficients DCT basse fréquence.

Nous avons proposé une organisation des données, en utilisant le plan comme unité de reconnaissance, basée sur une table de hachage, pour la recherche rapide de signatures. Plusieurs méthodes d'organisation des signatures dans la table, et plusieurs fonctions hachage sont proposées.

L'organisation des données proposée, permet de retrouver extrêmement rapidement un plan candidat, à partir d'un plan requête. Un des points critique de la méthode est alors la définition d'une distance entre plans, qui soit à la fois robuste aux problèmes d'alignement, et suffisament rapide pour ne pas perdre le bénéfice de l'indexation directe. Nous avons proposé une méthode très simple qui utilise l'information de position de la signature comme une *ancre* pour aligner la requête et le candidat.

Les résultats sont tout à fait corrects en terme de qualité, même si des difficultés apparaissent lorsque les plans ont des contenus mal représentés par la signature (peu d'information, très fort mouvement). Ces difficultés pénalisent essentiellement le rappel,

qui peut toutefois être amélioré lorsque l'EVR est redondant, c'est à dire que plusieurs répétitions d'un même plan sont présentes dans l'EVR. La précision est, par contre, excellente, avec des fausses alarmes présentes seulement de manière exceptionnelle.

Enfin, la méthode est capable de gérer des EVR de taille assez importante, avec des temps de recherche extrêmements rapides : seulement un peu plus d'une seconde est nécessaire pour détecter les plans communs entre une vidéo requête de 24 heures et un EVR de 500 heures.

Chapitre 3

Segmentation du flux en programmes

3.1 Stratégie

La segmentation du flux en programmes[1] est la détection et la délimitation des programmes dans un flux télévisé.

Nous définissons, tout d'abord, un *inter-programme* (en abrégé IP) comme une partie du flux qui respecte l'une ou l'autre de ces deux propriétés :

- est une publicité, selon la définition du CSA[sdl], c'est à dire « toute forme de message télévisé diffusé contre rémunération ou autre contrepartie en vue soit de promouvoir la fourniture de biens et de services, soit d'assurer la promotion commerciale d'une entreprise publique ou privée. ». Les émissions de type téléachat sont exclues de cette définition.
- fait partie de l'habillage de la chaîne (ex. jingles).
- est de l'autopromotion (ex. bande annonce).

Les inter-programmes sont plus précisement répartis en quatre types : les publicités, les parrainages, les bandes annonces, et les jingles. Ces derniers font, en général, partie de l'habillage de la chaîne. Les programmes sont définis à contrario, comme étant l'ensemble des parties du flux qui ne sont pas des inter-programmes. L'ensemble des programmes et des inter-programmes forment une partition du flux.

Le problème de la segmentation du flux en programmes peut être vu de plusieurs façons. La première est de chercher à détecter les frontières programmes/inter-programmes, à la manière d'une segmentation en plan par exemple. Cette méthode semble peu réaliste, sachant que la frontière est essentiellement une rupture sémantique, et est donc très difficilement détectable automatiquement, sauf cas particulier. La deuxième façon est de détecter directement les segments de programmes, ou d'inter-programmes. Il y a donc encore deux cas.

[1] appelée parfois en raccourci segmentation P/IP, c'est à dire segmentation en programmes/inter-programmes.

La détection des programmes : bien que plus intuitive car plus proche du fonctionnement humain, cette approche parait difficile : les programmes n'exhibent aucune caractéristique commune qu'il serait possible de détecter. Liang et al. [LLXT05] proposent bien une approche basée sur les génériques de début de programmes, mais cette approche n'est pas possible sur les chaînes françaises.

La détection des inter-programmes : détecter les programmes *a contrario* en détectant les inter-programmes est bien plus facile. L'état de l'art a montré que de nombreuses méthodes existent pour détecter les publicités. Cependant, les publicités ne forment qu'une partie des inter-programmes, et cette méthode fait de plus l'hypothèse qu'il existe des inter-programmes, ou du moins une transition entre chaque programme.

C'est l'approche par détection des inter-programmes qui parait la plus viable. Elle possède toutefois deux limitations. La première concerne l'extension des méthodes existantes aux inter-programmes et non aux seules publicités. L'état de l'art a montré que les méthodes à base de reconnaissance développées dans ce cadre pouvaient facilement s'étendre à la détection des IP. Il suffit en effet qu'ils soient dans la base de référence, mais la construction d'une telle base est alors un problème. La deuxième limitation est que, s'il n'existe pas d'inter-programmes entre deux programmes différents, alors la méthode produit un résultat erroné.

Avec ces limitations en tête, nous choisissons de détecter les inter-programmes par la méthode de reconnaissance exposée au chapitre précédent. Cette méthode ne permet cependant pas de détecter toutes les instances d'IP, en particulier, ceux qui n'ont jamais été diffusés, et ceux qui ne sont pas dans la base de référence. Nous proposons donc une méthode complémentaire, destinée à augmenter le rappel. Il s'agit d'une méthode tout à fait classique de détection d'instants d'apparition simultanée d'images monochromes et de silence. Ces instants sont diffusées en début et/ou fin de certains IP, ils sont appelés des **séparations**.

Cette méthode est tout à fait critiquable, notamment parce que tous les pays n'utilisent pas les séparations, en particulier les pays asiatiques [LQZ04], et parce que les séparations ne sont pas systématiques aux bornes des IP. Toutefois, c'est une méthode rapide et légère, qui fournit d'assez bons résultats (cf. section 3.2.3). De plus, l'ensemble des chaînes hertziennes françaises utilise ces séparations. Ces raisons nous ont semblées suffisantes pour utiliser les séparations comme information d'appoint de détection des IP. Les sections suivantes montrent comment les détecter, et étudient leur intérêt pour la détection des bornes des IP.

L'algorithme général de la segmentation en programme possède deux phases : une phase de détection et une phase de segmentation. La phase de détection comprend la détection des séparations, et la détection des plans répétés, ces détections se faisant indépendemment l'une de l'autre. La phase de segmentation consiste à utiliser les résultats de ces détections, qui permettent de créer une pré-segmentation. L'étape suivante consiste alors à classer les pré-segments en programme ou inter-programme, puis à les fusionner, ce qui donne la segmentation finale en programmes. Un schéma fonctionnel de la méthode est donné sur la figure 3.1.

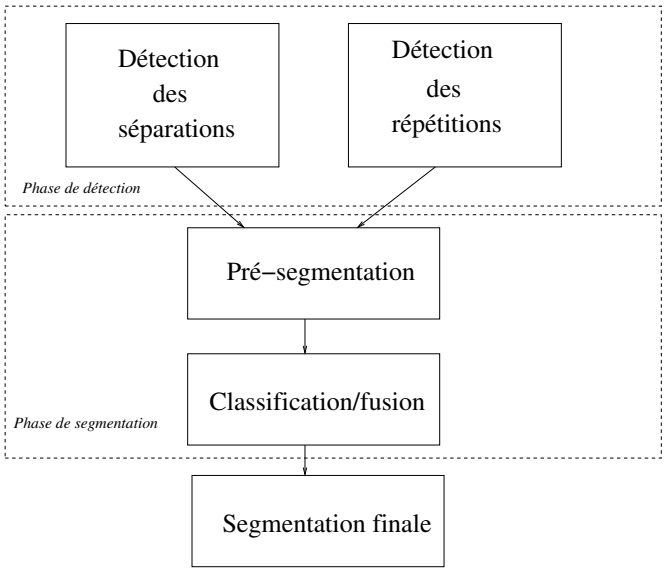

FIG. 3.1 – Principe général de la segmentation en programme.

3.2 Détection des séparations

Nous rappelons que nous appelons **séparations**, les séquences d'images monochromes accompagnées de silence, qui sont diffusées aux bornes des inter-programmes sur les chaînes françaises. Nous présentons de façon séparée la méthode de détection des images monochromes, la détection de silence, puis la méthode de fusion de ces deux résultats.

3.2.1 Détection des images monochromes

Malgré son apparente simplicité, ce problème est assez délicat, pour de multiples raisons :

- la couleur des images insérées dépend de la chaîne (noir, bleu, blanc), une même chaîne pouvant utiliser plusieurs couleurs d'images suivant le type de coupure qu'elle souhaite effectuer ;
- ces images sont très bruitées ;
- il y a des très nombreuses fausses alarmes (film très sombre, changement de scène dans les téléfilms...) ;
- un cadre peut entourer l'image : elle est alors bicolore.

Des méthodes simples pour détecter des images noires ont été proposées dès le début de travaux sur la détection de publicités [LKE97, SMOM02]. Ces méthodes ne sont toutefois pas assez robustes, ou ne traitent que le cas des images noires.

Nous proposons ici d'utiliser l'entropie de l'histogramme de luminance dans le domaine YUV. Pour un histogramme h quantifié sur N niveaux, son entropie H est donnée

par :

$$H = -\sum_{i=1}^{N} p_i \log p_i \qquad \text{avec } p_i = \frac{h(i)}{\sum_k h(k)}$$

En pratique, l'histogramme est quantifié sur $N = 48$ niveaux afin de réduire l'influence du bruit.

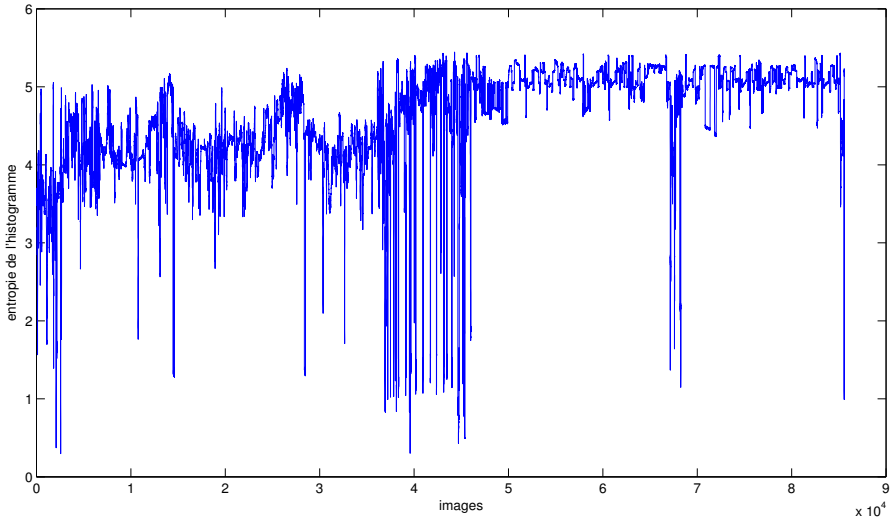

FIG. 3.2 – Variation de l'entropie de l'histogramme de luminance, sur 1h de télévision

H mesure la quantité d'information moyenne contenue dans l'histogramme. Une image parfaitement monochrome est d'entropie nulle, ce qui n'est bien évidemment pas le cas en pratique, mais la valeur de l'entropie reste tout de même très faible. Nous pouvons, de cette manière, détecter n'importe quelle image monochrome (blanche, noire...). L'intérêt de l'entropie est d'utiliser la totalité de l'information contenue dans l'histogramme, ce qui n'est pas le cas par exemple, d'une méthode qui utiliserait simplement le maximum de l'histogramme. L'entropie est aussi moins sensible à la quantification de l'histogramme qu'une méthode basée sur le maximum, et donne de meilleurs résultats, voir figure 3.7. La comparaison des résultats entre les deux caractéristiques proposées, le maximum, et l'entropie de l'histogramme de luminance, ne sont donnés qu'en section 3.2.3, afin d'évaluer les performances sur l'ensemble de la méthode de détection des séparations.

La figure 3.2 illustre la variation de l'entropie de l'histogramme sur 1h de la journée du 16/11/2004, sur la chaîne France2. Une longue plage de publicité est présente au milieu de la séquence. Le seuil de détection sur l'entropie est choisi à 2.

Toutefois, l'approche image est sujette à de nombreuses fausses alarmes, car de nombreuses images monochromes ne font pas partie d'une séparation. De plus, la présence d'une séparation n'est pas systématique aux bornes d'un IP. Afin d'avoir une idée de l'efficacité de la détection des séparations en tant que détecteurs d'IP, nous n'évaluons

donc pas la détection des séparations, qui n'est pas un but en soi, mais la détection des ruptures entre deux IP, et entre un programme et un IP. Certaines de ces ruptures ne sont pas signalées par des séparations. Le pourcentage de ruptures détectables à partir de la détection des séparations, est un indicateur de l'intérêt global de l'utilisation de la détection des séparations.

Le tableau 3.1 montre les résultats sur une séquence de télévision française d'une durée de 5h30. La ligne *détection* indique la précision et le rappel de la présence d'une rupture, tandis que la ligne *localisation* indique la précision et le rappel du nombre d'images détectées comme appartenant à une rupture. Les résultats ne sont pas très bons, puisque la précision est de seulement 41%, c'est à dire que de très nombreuses fausses alarmes sont présentes. On peut toutefois remarquer que la localisation est très bonne, avec une précision de 96%.

	Précision	Rappel
Détection	0.41	0.89
Localisation	0.96	0.79

TAB. 3.1 – Détection des ruptures - image seule.

3.2.2 Détection de silence

3.2.2.1 Rappels de traitement audio

Nous donnons, tout d'abord, quelques définitions de base concernant le traitement du signal audio. Nous nous inspirons ici fortement de Pinquier [Pin04] pour les définitions.

Un signal audio numérique est un signal 1D représenté par une suite d'échantillons. Le nombre d'échantillons par seconde dépend de la fréquence d'échantillonage du signal. Des valeurs typiques sont de 16, 32, 44.1 et 48 kHz. Les traitements sont effectuées sur une *trame*, qui est une suite d'échantillons sonores de l'ordre de 10 à 40 ms. Un signal audio étant, en général, non-stationnaire, le découpage en trames permet de le considérer comme stationnaire sur chaque trame et donc d'appliquer les méthodes classiques de traitement du signal. Les trames possèdent généralement un recouvrement, partiel ou total, qui permet d'éviter les effets de bords entre deux trames consécutives.

Si l'on désigne par $x_n(i)$ le $n^{\text{ième}}$ échantillon de la trame i, et N le nombre d'échantillons dans la trame, alors on définit l'énergie d'une trame par :

$$E(i) = \sum_{n=1}^{N} x_n^2(i)$$

qui est, en général, exprimée en décibel :

$$E_{db}(i) = 10 \log_{10} \sum_{n=1}^{N} x_n^2(i)$$

L'énergie peut aussi être normalisée par rapport à son maximum observé. La figure 3.3 montre l'énergie normalisée sur un signal audio provenant de la télévision. Le signal

est extrait d'un téléfilm lors d'une scène de dialogue, et est donc essentiellement de la parole, plus une musique de fond.

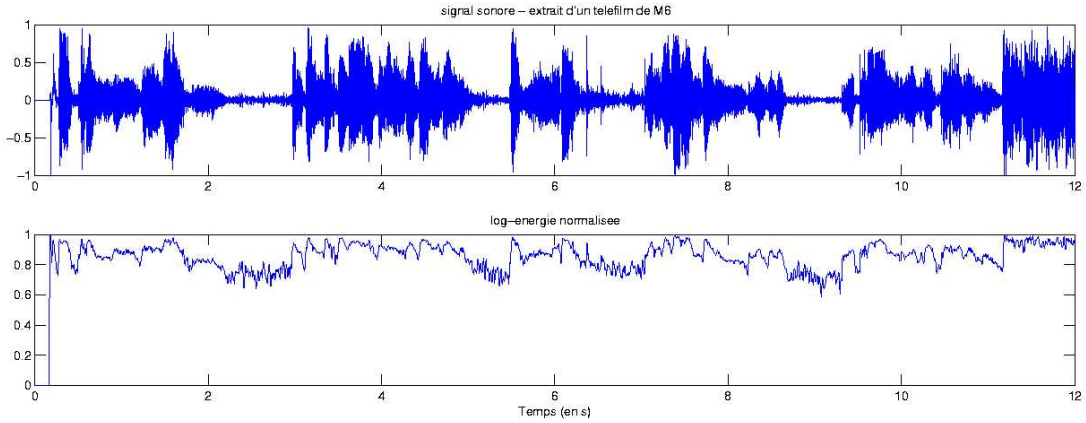

FIG. 3.3 – Signal extrait d'un téléfilm de la chaîne M6 et son énergie normalisée.

Une autre caractéristique utile dans le cadre de la détection de silence est le ZCR, *Zero Crossing Rate*, qui est le nombre de fois où le signal change de signe sur une trame. Il est défini par :

$$ZCR(i) = \frac{1}{2N}\left(\sum_{n=1}^{N} |sign(x_n(i)) - sign(x_{n-1}(i))|\right)$$

3.2.2.2 Méthode de détection de silence

La détection de silence est un sujet relativement simple et bien étudié par la communauté audio. La plupart des méthodes proposées se base sur l'énergie du signal. La plus simple est tout simplement de seuiller l'énergie. Toutefois, cette approche n'est généralement pas robuste, et génère de nombreuses fausses alarmes sur un signal bruité. Une approche un peu plus robuste est proposée par Harb et al. [HCA01], qui utilise l'énergie normalisée et le ZCR pour détecter les segments de silence. Les deux caractéristiques sont multipliées et le produit est alors seuillé.

Une autre approche proposée par l'équipe METISS de l'IRISA consiste à modéliser l'énergie du signal par un modèle bi-gaussien. La gaussienne de plus petite moyenne représente le silence tandis que la deuxième correspond à de l'activité audio. Deux méthodes sont proposées pour réaliser l'étiquetage d'un segment en tant que silence. La première se base sur un critère de maximum de vraisemblance, en estimant quel est le modèle qui donne la meilleure vraisemblance. La deuxième méthode prend une décision en seuillant la moyenne de l'énergie du segment. Le seuil est fixé par $seuil = m_h - a\sigma_h$, avec (m_h, σ_h) les paramètres de la gaussienne modélisant les hautes énergies, et a le paramètre règlant l'éloignement entre le seuil et la moyenne m_h. Ce seuil permet de différencier les deux gaussiennes, comme illustré sur la figure 3.4. La méthode permet

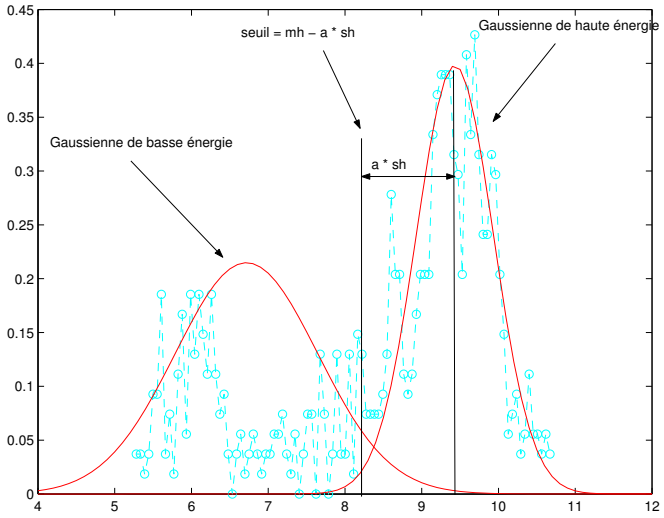

FIG. 3.4 – Illustration du modèle bi-gaussien, et de la valeur du seuil.

aussi de ne conserver que des segments d'une longueur minimale donnée. La première méthode est évidemment la plus intéressante, puisqu'elle permet d'éviter de recourir à un seuil arbitraire, et se base uniquement sur la modélisation du signal en terme de bi-gaussiennes.

Nous avons essayé trois méthodes de détection différentes. Une première méthode très simple, basée sur un seuillage de l'énergie, et qui ne retient que les segments d'une durée minimale donnée (30 ms). La deuxième méthode est celle de l'équipe Metiss, basée sur le maximum de vraisemblance, et la troisième est celle basée sur le seuillage de l'énergie moyenne, toutes deux avec le même paramètre de longueur de segment de silence minimum de 30 ms.

La première méthode donne des résultats excellents, donnés sur la figure 3.2, et est, de plus, de faible complexité. La deuxième méthode, en revanche, produit des résultats n'ayant pas de sens, aucune séparation n'étant correctement trouvée. Cette méthode est, en fait, non adaptée à nos signaux et à notre problématique. La figure 3.5 montre la variation de l'énergie sur quelques dizaines de minutes en présence de séparations et montre que l'on observe une **coupure du signal** lors des séparations. Ce phénomène a été observé sur l'ensemble des chaînes hertziennes françaises, et est aussi observable sur une durée plus importante, sur une chaine différente, sur la figure 3.6. On peut alors supposer que la méthode estime mal la gaussienne de basse énergie, car le nombre d'échantillons appartenant à cette classe est très faible.

La troisième méthode nécessite la détermination d'une valeur du seuil a, qui puisse donner de bons résultats. Cette valeur doit être choisie très élevée pour obtenir des résultats satisfaisants, et est alors inférieure à m_l, la moyenne de la gaussienne de basse énergie. Ceci n'a plus vraiment de sens, et ramène la méthode à un simple seuillage de l'énergie trame par trame, ce qui est équivalent à la première méthode. En pratique la

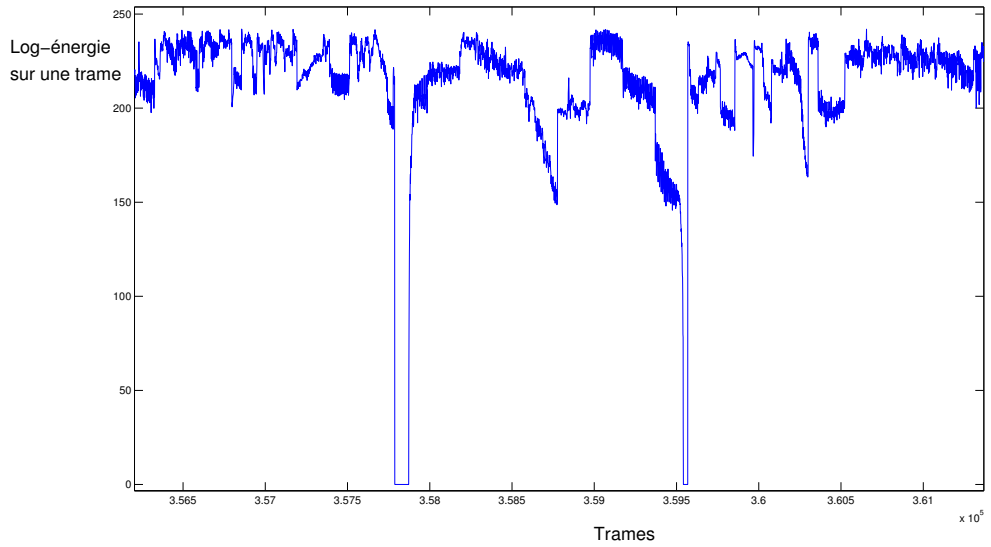

FIG. 3.5 – Variation de l'énergie du signal audio en présence de séparations sur la chaîne France2.

première et la troisième méthode donnent, effectivement, les mêmes résultats.

Nous choisissons de travailler avec la première méthode, d'une part parce que la modélisation bi-gaussienne échoue, et d'autre part pour des raisons pratiques. La troisième méthode doit, au préalable, effectuer la modélisation en calculant les paramètres des deux gaussiennes. Ceci nécessitait donc le parcours du signal entier, ce qui, sur des signaux de 24h, n'était pas possible, pour des raisons de complexité mémoire. La méthode nécessitait, de plus, le décodage complet du fichier avant tout calcul, alors que la première méthode peut s'effectuer à la volée.

Pour le calcul de l'énergie, des trames de 10 ms avec un recouvrement de 50% ont été utilisées. Le seuil pour la détection est fixé à 60, et seuls les segments de plus de 30 ms sont conservés. Les résultats sont présentés dans le tableau 3.2 sur la même séquence de télévision de 5h30 que pour l'image. Tout comme pour la détection des images monochromes, nous évaluons non pas la détection des séparations, qui n'est pas un but en soi, mais la détection des ruptures entre deux IP, ou entre un programme et un IP. De même que pour les images monochromes, une rupture n'est pas forcément signalée par une séparation audio.

	Précision	Rappel
Détection	0.82	0.9
Localisation	0.84	0.79

TAB. 3.2 – Détection des ruptures - audio seul

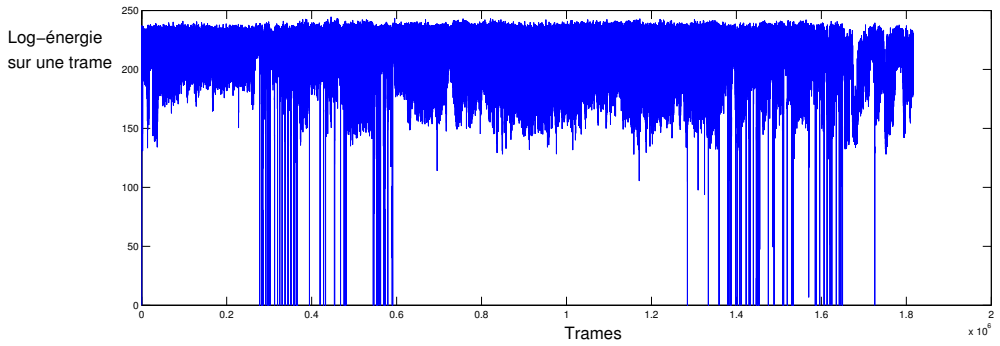

FIG. 3.6 – Variation de l'énergie du signal audio en présence de séparations sur la chaîne TF1.

Ces résultats montrent que l'énergie du signal audio est un plutôt un bon indicateur, notamment pour ce qui est de la précision, qui est bien meilleure que pour l'image. Les fausses alarmes sont dues à des changements de scènes dans certains téléfilms et documentaires, ou parfois à la coupure entre plateau/reportage dans les journaux d'information.

Enfin, les figures 3.6 et 3.5 permettent de faire une remarque annexe, sans rapport avec la détection de silence. Elles montrent que l'énergie ne subit pas particulièrement d'augmentation pendant un inter-programme, ce qui confirme la remarque du paragraphe 1.2.1, page 26, sur le volume des publicités à la télévision.

3.2.3 Fusion audiovisuelle

Il semble raisonable de penser que l'utilisation de plusieurs médias peut améliorer les résultats de détection. L'intégration de différentes sources d'information pose cependant un problème de décision lorsque les résultats sont contradictoires. De plus, l'audio et la vidéo ont des fréquences d'échantillonnage différentes, il y a donc aussi un problème de synchronisation.

On distingue généralement trois méthodes de fusion des informations multimodales : l'intégration précoce, l'intégration tardive, et l'analyse successive. L'intégration précoce consiste à intégrer les attributs audios et vidéos au sein d'un même vecteur avant la classsification. L'intégration tardive consiste à classifier indépendemment les portions du flux pour chaque modalité, puis à combiner les résultats de ces classifications. L'approche par analyse successive consiste à analyser tout d'abord le flux avec une seule modalité, puis à analyser les portions du flux détectées par la seconde modalité. Cette approche est la plus intéressante lorsque les modalités sont clairement complémentaires. C'est le cas ici, où le détecteur de silence exhibe de très bonnes performances en terme de détection, tandis que la méthode basée image permet de localiser finement les bornes de la séparation.

La méthode consiste donc, tout d'abord, à repérer les segments de silence à l'aide de

la technique exposée en 3.2.2, ce qui nous donne un intervalle de silence $I_1 = \{i_0 \dots i_n\}$. Une décision basée sur l'attribut image dans l'intervalle I_1 ne donnerait pas de très bon résultats, car la localisation de la séparation par la méthode basée audio n'est pas très bonne. Cette localisation est au contraire excellente par la méthode basée image. L'attribut image est donc utilisé pour détecter précisement les limites de la séparation dans un intervalle élargi $I_2 = \{i_0 - \gamma, \dots, i_n + \gamma\}$

L'agorithme est le suivant. On parcourt l'intervalle I_2. Un début de segment est détecté dès que $H(i) \leq \alpha$. Le segment se termine lorsque $i = i_n + \gamma$ ou que le nombre maximal d'exceptions est atteint. On autorise en effet k exceptions $\{j_1 \dots j_k\}$ telles que $\forall p \in \{1 \dots k\}$, $H(j_p) > \alpha$. En pratique, $k = 2$, $\gamma = 10$.

	Détection		Localisation	
Modalité	Precision	Rappel	Precision	Rappel
Audio	0.82	0.9	0.84	0.79
Image	0.41	0.89	0.96	0.79
Fusion	1	0.9	0.94	0.74

TAB. 3.3 – Détection des ruptures - son et image

Les résultats de la fusion des modalités sont donnés dans le tableau 3.3. Le principe même de la méthode fait que le rappel de la fusion ne peut excéder celui de l'audio. Par contre la précision est très nettement améliorée puisqu'il n'y a pas de fausses alarmes, et ceci sur 5h30 de télévision. Des fausses alarmes ont toutefois étés constatées sur d'autres vidéos pour lesquelles nous ne disposions pas de vérité terrain. Ces fausses alarmes sont principalement dues aux changements de scènes dans certains téléfilms. De façon plus marginale, la vérité terrain indique des intervalles assez larges pour les changements, d'où les mauvais scores de rappel pour la localisation.

Nous donnons aussi, à titre indicatif, sur la figure 3.7, les résultats comparatifs de la méthode de fusion, pour les deux méthodes de détection évoquées en section 3.2.1, qui diffèrent simplement par la caractéristique utilisée, le maximum, ou l'entropie de l'histogramme de luminance de l'image. Les résultats sont donnés sous la forme de courbes précision/rappel, où le paramètre variable est le seuil α. Les résultats sont clairement en faveur de l'entropie.

3.3 Détection des répétitions

Le détail de la méthode de la détection des répétitions a été présenté au chapitre 2.

L'utilisation d'une méthode de reconnaissance nécessite un ensemble de vidéos de référence (appelé EVR) dans lequel sont contenus les segments à reconnaitre. Dans le cas des méthodes classiques de détection de publicité, vues en 1.2, les EVR utilisés ne comprennent que des IP. Une détection signifie alors que le segment reconnu est un IP. Dans notre cas, les segments contenus dans l'EVR sont des plans, et peuvent être aussi bien des plans de programmes que des plans d'inter-programmes. La raison de la présence de plans de programmes dans l'EVR est que certains d'entre eux (génériques,

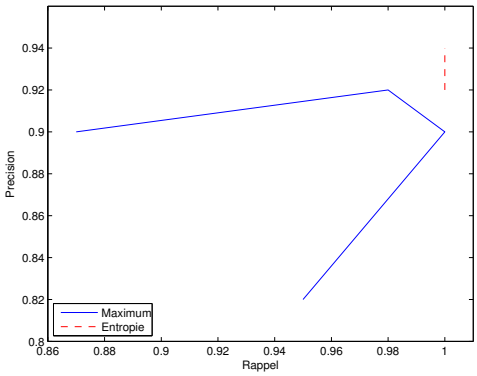

FIG. 3.7 – Comparaison des deux méthodes basées sur l'histogramme de luminance de l'image, sur une séquence d'une heure.

jingles, séquences de présentation, etc...) se répètent régulièrement, et possèdent donc aussi un intérêt pour la structuration. En conséquence, chaque plan de l'EVR se doit d'être étiqueté par *type* (P ou IP), puisque sans cette information, une reconnaissance n'apporterait aucune information utile sur la nature du plan reconnu.

L'EVR est une collection de programmes et d'inter-programmes. Les programmes sont étiquetés par leur nom, manuellement, à partir du guide des programmes si l'information est disponible, à partir du texte affiché à l'écran sinon. Les IP sont divisés en quatre sous-type (parrainage, publicité, bande annonce, et jingle), et ils sont aussi étiquetés par un nom, sauf les publicités. Cette distinction en différents types d'IP, et le fait de les nommer n'est pas directement utile à la structuration, sauf en ce qui concerne les bandes annonces, comme nous le verrons dans le paragraphe 3.4. À titre d'information, on peut voir sur la figure 3.8, un exemple d'étiquetage manuel avec les différents sous-types d'IP.

L'EVR utilisé est **statique**, c'est à dire qu'il n'est pas mis à jour au fil du temps, et qu'il est identique quelque soit la vidéo à structurer. Ici, l'EVR est constitué de 24 heures de télévision, enregistrées en continu le 9/05/2005 sur la chaîne France2. Son étiquetage a été réalisé manuellement, par nos soins.

Nous définissons maintenant la notion d'*EVR complet*. On appelle un EVR complet pour une vidéo, un EVR qui contient l'ensemble des inter-programmes de cette vidéo. De manière plus générale, la complétude d'un EVR par rapport à une vidéo est définie comme le pourcentage de plans d'inter-programmes de la vidéo qui sont présent dans l'EVR. Un EVR complet a donc une complétude de 100%. Nous verrons dans les résultats, en section 3.4.2.4, ainsi que dans le chapitre 5, l'importance de cette notion pour la qualité de la segmentation.

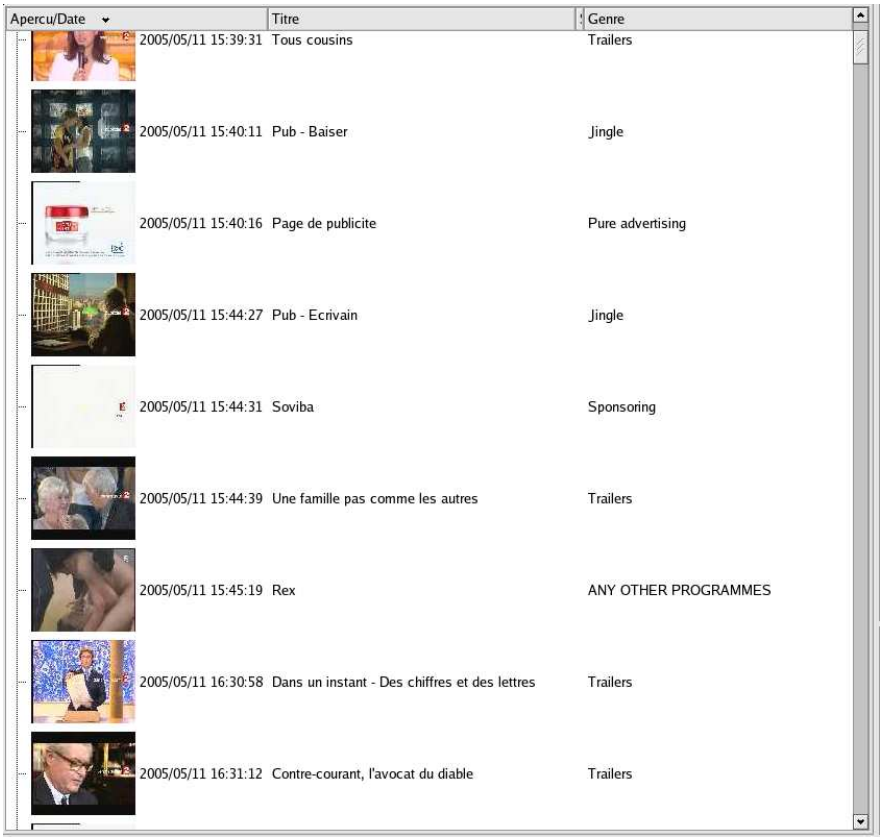

FIG. 3.8 – Exemple d'étiquetage manuel sur quelques heures de la chaîne France2

3.4 Segmentation en programmes

3.4.1 Méthode

La segmentation du flux en programmes se fait en trois étapes : une étape de pré-segmentation, une étape de classification, et une étape de fusion. Ces trois étapes sont illustrées sur la figure 3.9.

La pré-segmentation consiste à soustraire du flux les images détectées comme étant des inter-programmes par les méthodes précédentes. Si l'on considère l'ensemble F des images du flux, S l'ensemble des images ayant été détectées comme des séparations, et R l'ensemble des images ayant été reconnues comme des inter-programmes, alors nous pouvons définir trois types de pré-segmentation PS :

- pré-segmentation utilisant les séparations seules : $PS_1 = F \setminus S$
- pré-segmentation utilisant les segments répétés seuls : $PS_2 = F \setminus R$
- pré-segmentation utilisant les séparations et les IP répétés : $PS_3 = F \setminus (R \cup S)$

Le flux est ainsi découpé en un ensemble de pré-segments, visibles sur la figure 3.9.

La classification consiste à classer les pré-segments, en programme ou inter-programme, suivant leur longueur. De manière très simple, on choisit une longueur maximale, que l'on nomme le **seuil de classification P/IP**, pour laquelle tout pré-segment de longueur inférieure est un pré-segment d'IP, et inversement, tout pré-segment de longueur supérieure est un pré-segment de programme.

La fusion est effectuée à la suite de cette classification. Elle consiste à fusionner ensemble les pré-segments d'IP, les séparations, et les IP répétés, contigus. Lorsqu'un pré-segment d'IP est adjacent à un IP répété, et/ou à une séparation, alors ils sont fusionnés en un seul **segment d'IP**. Les séparations, ou les IP répétés, isolés forment aussi des segments d'inter-programmes. Enfin, les pré-segments classés comme étant des programmes deviennent des **segments** de programmes. Nous emploierons désormais le terme segment (sous entendu de programme ou d'inter-programme) comme étant les éléments résultants de ce processus.

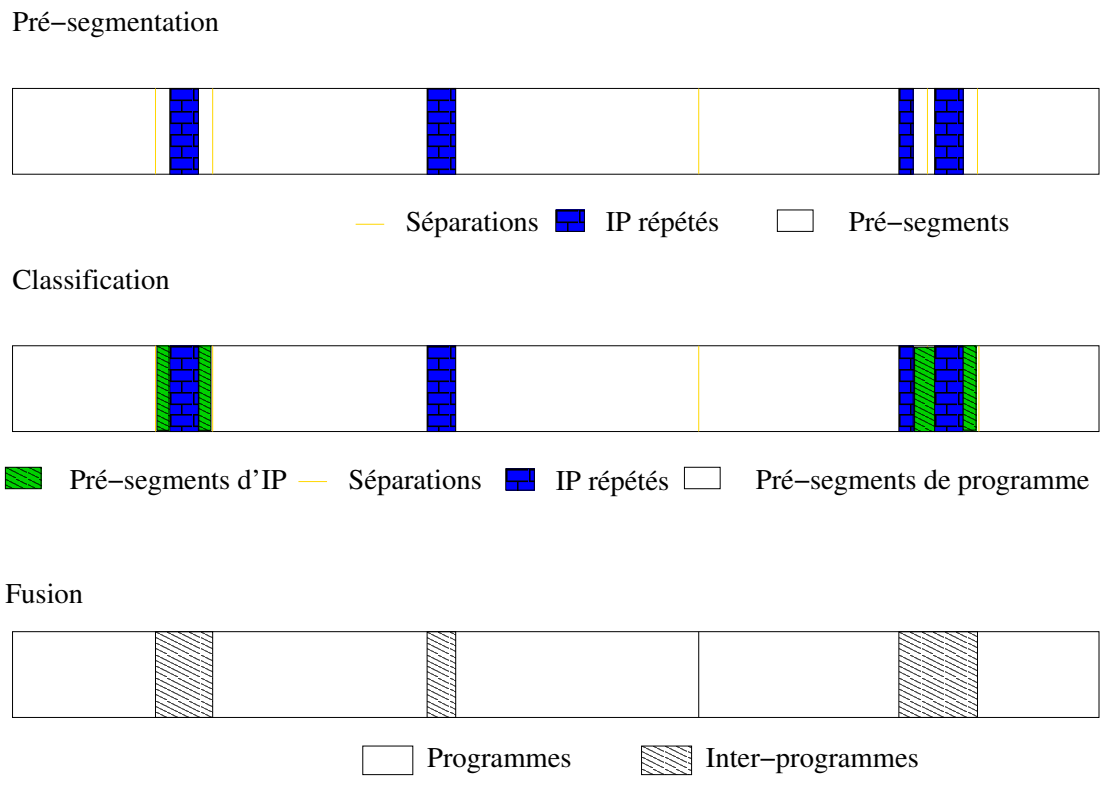

FIG. 3.9 – Principe de la segmentation en programme : détection, pré-segmentation, puis classification

La méthode de classification peut paraitre un peu simpliste, mais la longueur d'un pré-segment est la seule information dont nous disposons pour faire la distinction P/IP. Notons que c'est aussi la méthode employée par [CBF06] et [Her05]. La valeur du

seuil de classification P/IP est cruciale, car une valeur trop grande ne permettra pas d'identifier les plages de programmes les plus courtes, et une valeur trop petite produira des mauvaises classifications de périodes d'inter-programmes en tant que programmes. De plus la meilleure valeur de ce seuil dépend de la qualité de la segmentation. Ainsi avec un EVR *complet*, c'est à dire qui détecte l'ensemble des IP, le seuil de classification P/IP peut être assez bas, car il y a alors peu de chances qu'un IP soit classifié en tant que programme. La valeur finalement retenue a été déduite à partir de la vérité terrain, en choisissant celle qui maximise la F-mesure sur plusieurs segmentations de journées entières, soit une valeur d'environ $50s$.

La classification est un point délicat, car certains IP, en particulier les bandes annonces, sont assez longs (jusqu'à 40s voire 60s), et certains programmes peuvent être très courts (à peine 1 minute). Des erreurs de classification sont donc fréquentes en cas d'EVR incomplet. Il suffit, en effet, d'avoir un espace vierge de toute détection, (séparation ou IP répété), de plus de 50s, pour qu'il soit alors classé comme un segment de programme.

À cela s'ajoute le cas particulier des bandes annonces contenues dans l'EVR. Leur présence est extrêmement problématique. La raison en est que les plans diffusés dans une bande annonce le sont aussi dans le programme annoncé. Certaines parties du programme annoncé par la bande annonce sont donc reconnues comme étant un IP et segmentent donc le programme en plusieurs parties, produisant une sur-segmentation. L'approche choisie pour résoudre ce problème est d'utiliser le guide des programmes. Il suffit que les IP répétés qui portent le même nom qu'un programme adjacent dans le guide n'aient pas d'effet segmentant. Il est donc important pour la structuration que les bandes annonces soient étiquetées correctement.

3.4.2 Résultats

Les résultats sont calculés indépendamment sur 20 jours du corpus 2, du 10 au 30 mai 2005, soit une base de test d'un total de 480h de vidéo. Nous rappelons que le corpus est présenté en annexe A. L'EVR utilisé pour la détection des répétitions est constitué de l'ensemble de la journée du 9 mai 2005 de France2, étiquetée manuellement.

Bien que nous ayons présenté le problème comme un problème de segmentation, il est plus facile pour l'évaluation de le penser comme un problème de classification binaire (P ou IP), et d'utiliser les mesures classiques de précision et rappel, l'unité de classification étant l'image. Toutefois, le fait d'utiliser l'image comme unité de classification, c'est à dire que les résultats sont exprimés de manière temporelle, biaise la précision et le rappel de la segmentation par programme. En effet, les programmes composent la majorité du flux, il suffit donc d'indiquer une segmentation composée uniquement de programmes pour avoir des scores élevés de segmentation en programme, alors que la segmentation n'a aucun sens. L'interprétation de la mesure est toutefois intuitive puisqu'elle mesure le nombre d'images bien classées en tant que P ou IP.

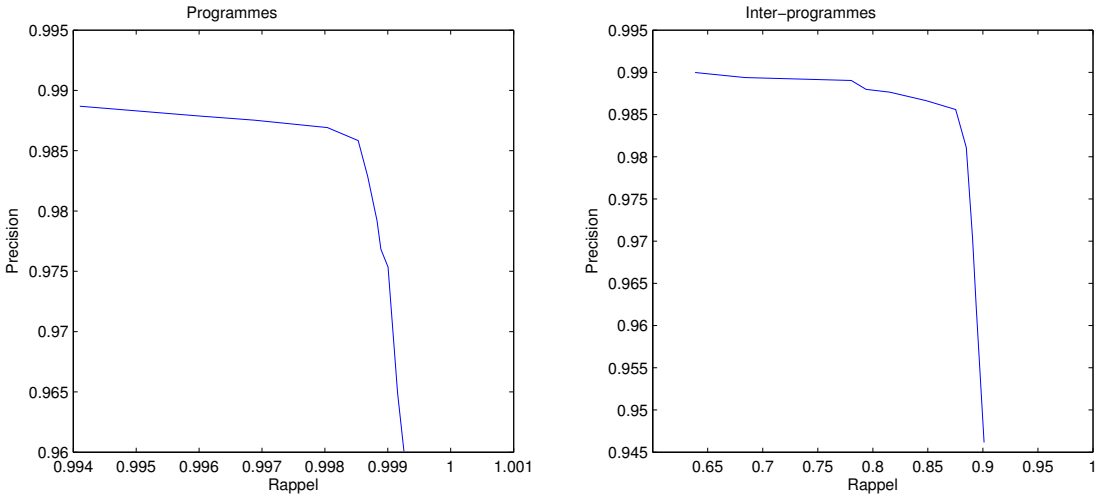

FIG. 3.10 – Segmentation P/IP moyenne sur 20 jours de télévision (480 heures), en fonction du seuil de classification P/IP.

3.4.2.1 Résultats en fonction du seuil de classification

Cette section étudie l'impact du seuil de classification sur les résultats. La section 3.4.1 avait annoncé que le choix de ce seuil était crucial. Nous rappelons que ce seuil est celui qui permet de classer les pré-segments en tant que programme ou inter-programme, suivant leur longueur. Nous cherchons donc à faire varier ce seuil et à déterminer l'impact sur les résultats en terme de classification P/IP. La figure 3.10 donne les résultats en terme de précision et rappel, moyennés sur les 20 jours de test, avec une variation du seuil de classification de 600 à 1600 images (24 à 64 secondes), par pas de 100 images. La figure montre effectivement que la variation du seuil de classification a un fort impact sur les résultats, et que ce seuil permet de contrôler la précision et la rappel.

Les scores très élevés de la segmentation en programme (99% en rappel et précision) sont à interpréter à la lumière de la remarque de la section précédente, concernant le biais dans les résultats. Le flux comprenant environ 88% de programmes, les mesures les plus intéressantes sont le rappel et la précision des inter-programmes.

Comme annoncé au paragraphe 3.4.1, la valeur choisie est celle qui maximise la F-mesure, soit une valeur de 1200 images, utilisée dorénavent comme valeur par défaut dans l'ensemble de la thèse.

3.4.2.2 Résultats en fonction de la méthode de segmentation

La table 3.4 donne la précision et le rappel de la classification en P/IP, moyennés sur les 20 jours de test, en utilisant un seuil de classification en P/IP de 1200 images. Les résultats sont donnés sous trois formes différentes : en utilisant les séparations seules, les répétitions seules, et en utilisant la combinaison des deux, selon la méthode expliquée

en 3.4.1. À des fins de comparaison, nous donnons aussi le score obtenu par le guide des programmes, qui nous sert de référence. Le guide n'indiquant que les programmes, la précision par programme est donc en fait le ratio P/(P+IP), c'est à dire, au vu des résultats du guide, qu'une journée est constituée en moyenne de 88.5% de programmes et 11.5% d'inter-programmes.

Méthode	Programme		Inter-programme	
	Précision	Rappel	Précision	Rappel
Guide des programmes	88.5	100	0	0
Séparations seules	97.4	99.4	93.5	75.1
Répétitions seules	95.2	99.9	99.2	54.1
Séparations+répétitions	98.5	99.9	98.7	85.9

TAB. 3.4 – Segmentation P/IP sur 20 jours de télévision (480 heures).

On voit à partir de ces résultats qu'une segmentation P/IP à partir des séparations seules est possible, mais a tendance à manquer un nombre important d'IP. Les répétitions seules ont aussi une excellente précision, mais un rappel encore plus faible. Les chiffres concernant les répétitions seules cachent en fait une grande disparité : les résultats sont en fait excellents pour les journées proches temporellement de la journée du 9 mai, qui nous sert d'EVR, et beaucoup moins bons en ce qui concerne les journées plus éloignées. La méthode qui combine séparations et répétitions génère par contre des résultats très corrects, qui cachent aussi une grande disparité suivant la proximité de la journée testée avec l'EVR. À titre indicatif, Les résultats avec la méthode combinée sont supérieurs à 92% de rappel en IP sur les trois premiers jours.

3.4.2.3 Résultats en fonction du temps

La baisse des résultats en fonction du temps est visible sur la figure 3.11. Elle montre les résultats jour par jour sur l'ensemble de notre corpus de 20 jours de France 2, en utilisant la F-mesure, définie par $F = \frac{2PR}{P+R}$ où P et R sont respectivement la précision et le rappel de la segmentation en IP. On peut y voir que les résultats décroissent avec le temps. Ceci est dû au vieillissement de l'EVR, qui n'est pas mis à jour.

La qualité de la segmentation est aussi particulièrement variable selon les jours, ce qui est dû à la variabilité du contenu (évenement particulier, apparition de bandes annonces, etc...), ou simplement au fait que l'EVR n'est pas assez complet pour générer une segmentation correcte. En particulier, on peut observer sur la figure 3.11, une forte baisse des résultats pour les journées du 24 et 26. Ceci n'est pas dû à un évènement particulier, mais au fait que l'EVR est de moins en moins complet au fil du temps, et qu'une seule non-détection d'IP peut avoir un très fort impact sur la segmentation. La figure 3.12 donne un exemple où une non-détection d'un seul plan (en bleu sur la figure) peut avoir un fort impact. Dans l'exemple 1, les pré-segments p2 et p3 sont classés comme IP du fait de leur petite taille, ce qui, par fusion, permet de former un large segment d'IP. Dans l'exemple 2, la simple disparition du plan reconnu comme étant un IP produit un pré-segment p2 large, qui est alors classé comme un pré-segment

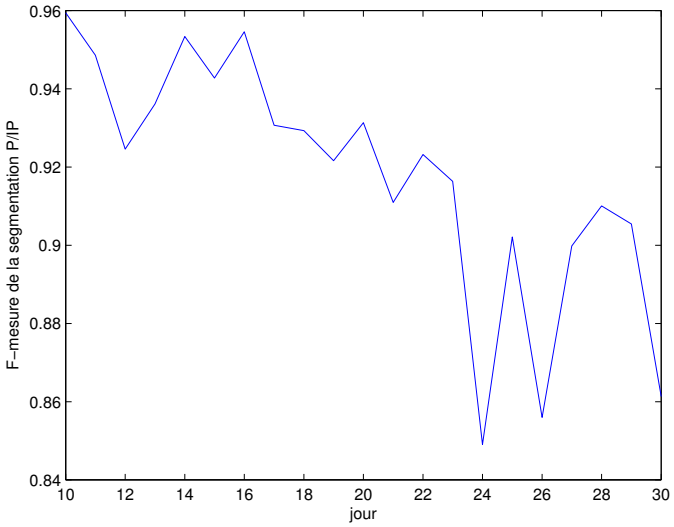

FIG. 3.11 – Segmentation en IP jour par jour sur 20 jours de télévision (480 heures).

de programme. L'impact sur la segmentation finale est alors important, puisque qu'un large segment d'IP est à tort classé comme un segment de programme.

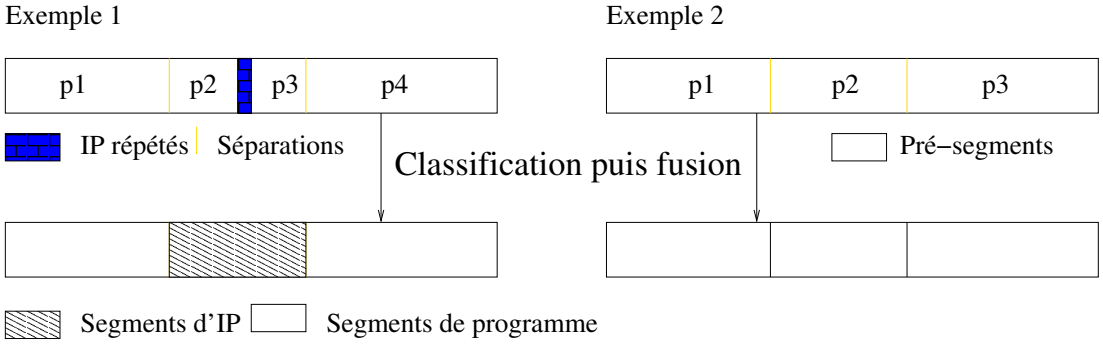

FIG. 3.12 – Exemple de non-détection d'IP avec une forte influence sur la segmentation.

3.4.2.4 Influence de la complétude de l'EVR sur la segmentation

Il semble évident que la complétude de l'EVR influe les résultats. Plus le pourcentage d'IP de la requête présents dans l'EVR est élevé, plus la segmentation sera précise. Il est cependant difficile de construire un EVR complet, puisque cela impliquerait de connaitre à l'avance tous les IP, ce qui n'est évidemment pas réaliste.

Afin de mesurer l'influence de la complétude de l'EVR sur la qualité de la segmen-

tation, nous reprenons les résultats de l'expérience précédente, où les résultats étaient donnés au cours du temps, avec un EVR *statique*, choisi comme étant la journée du 9/05/2005, c'est à dire la journée origine des 21 jours du corpus 2. Ce test est donc appelé « Origine ». La deuxième partie des résultats est obtenue est choisissant comme EVR la journée précédant la requête, par exemple pour segmenter la journée du 15/05, l'EVR utilisé est l'ensemble de la journée du 14/05. Ce test est appelé « Précédent », en référence à la position de l'EVR par rapport à la requête.

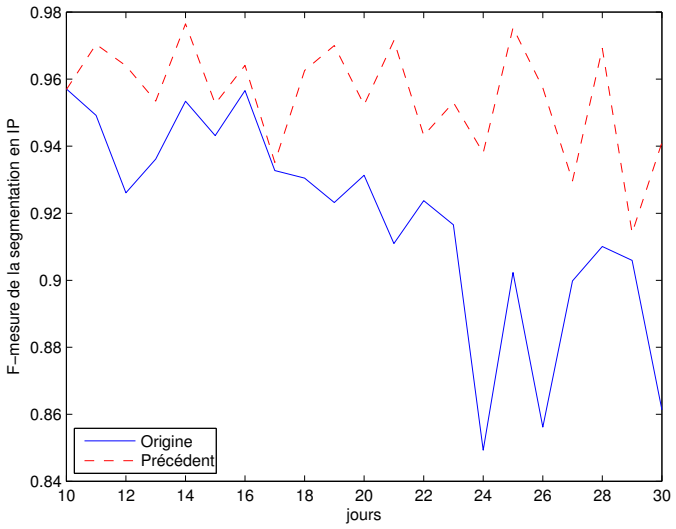

FIG. 3.13 – Influence de la complétude de l'EVR sur les résultats de segmentation en IP.

Les résultats sont donnés sur la figure 3.13. À noter que le cas du 10/05 est un cas particulier, puisque son EVR est le 9/05, et est donc identique au test « Origine ». Il est donc normal que les deux courbes partent du même point. Les résultats sont assez clairs, en dépit des variations quotidiennes importantes : au bout de 20 jours, la F-mesure a baissé d'environ 10% pour le test « Origine », alors que la F-mesure pour le test « Précédent » est restée stable.

3.4.2.5 Exemples de segmentation

Nous donnons dans cette section deux exemples précis qui illustrent les apports et les difficultés de la segmentation. Le premier est donné sur la figure 3.14, et illustre le gain apporté par la segmentation par rapport au guide des programmes. Les manques du guide sont, non seulement les bornes des programmes, mais aussi la non-indication de nombreux programmes courts. Il peut arriver que jusqu'à 5 programmes courts consécutifs ne soient pas annoncés dans le guide ! Dans cet exemple, la segmentation automatique colle parfaitement à la vérité terrain, alors que les indications du guide sont

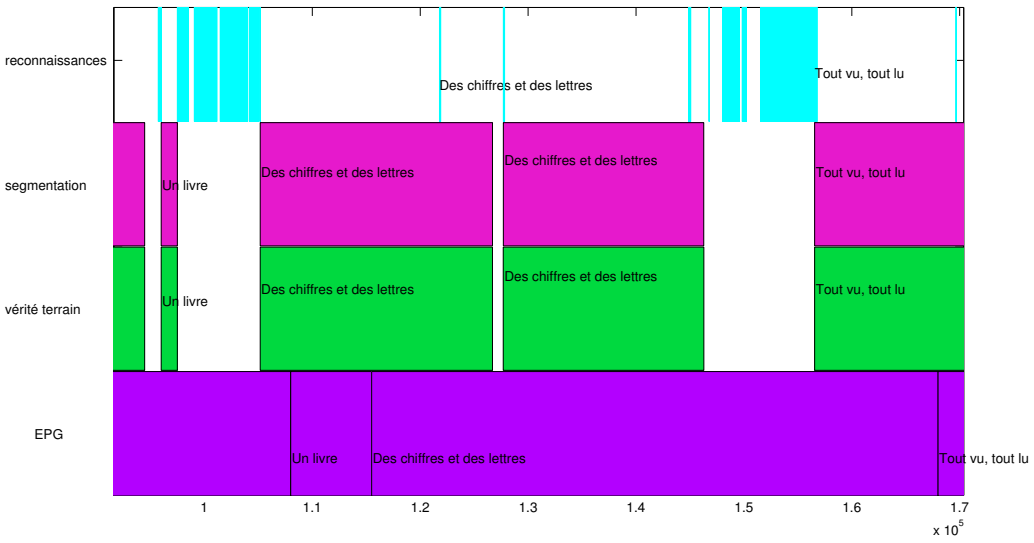

FIG. 3.14 – Exemple d'une segmentation d'un après-midi de la chaîne France2. Le guide est en violet, en bas, la vérité terrain en vert, la segmentation automatique est en rose foncé, et les reconnaissances sont indiquées en bleu clair.

franchement décalées.

Nous donnons aussi un contre-exemple dans la figure 3.15, où le guide des programmes est exceptionnellement précis[2] et où, au contraire, la segmentation est très mauvaise. Ce genre de situation est heureusement rarement aussi difficile, mais est caractéritique des diffusions de la nuit, où il y a peu de publicité et donc peu ou pas de coupures entre les émissions. Cette constatation nous ramène à l'hypothèse faite au début de ce chapitre, en 3.1, qui était que les programmes sont séparés par des inter-programmes, ou, au pire, par une séparation. Cette hypothèse est donc fausse pour les programmes diffusés la nuit, qui peuvent s'enchainer sans transition. Domenget [Dom00] cite aussi le cas pour des programmes de plus grande écoute, où l'absence de transition est volontaire, afin de renforcer l'impression de continuité du flux télévisuel. Ce procédé n'est pas présent dans notre corpus, mais s'il était amené à se généraliser, notre méthode n'aurait alors plus aucun intérêt au niveau de la structuration, se résumant à une méthode de détection de coupures intra-programmes.

[2]Le guide est pourtant particulièrement peu fiable la nuit, qui est le moment où les chaînes peuvent rattrapper le retard pris pendant la journée, et il est en général fréquent que les émissions soient supprimées, remplacées, ou interverties.

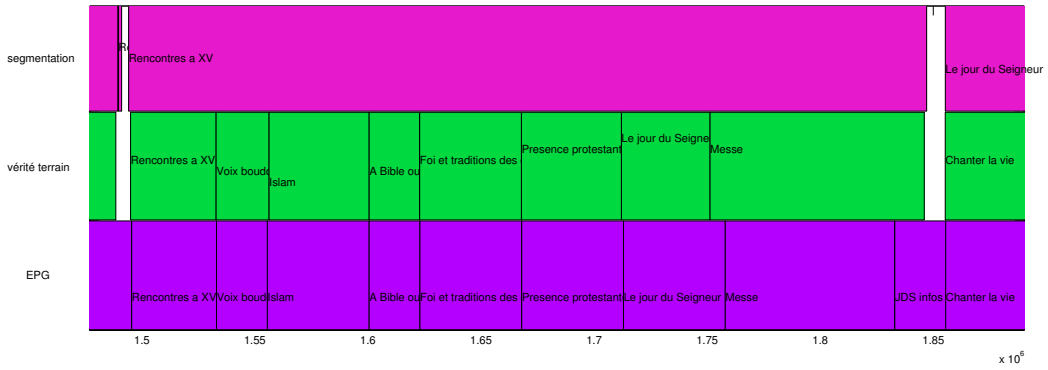

FIG. 3.15 – Exemple de segmentation problématique et de guide des programmes inhabituellement correct : la nuit du 30/05/2005 sur France2.

3.5 Synthèse

Ce chapitre a montré comment effectuer la segmentation d'un flux de télévision en programmes et inter-programmes. Cette segmentation est produite à partir de deux sources. La première est la détection des *séparations*, répérées par leur caractéristiques images et audio bien particulières. La deuxième est la reconnaissance des segments d'inter-programmes, grâce à un ensemble d'inter-programmes pré-étiquetés. L'ensemble des détections (séparations+répétitions) segmente le flux en un ensemble de présegments, qui sont ensuite classés en programmes ou inter-programmes à partir de leur longueur.

Les résultats sont très encourageants, produisant une excellente segmentation en journée. Des difficultés importantes apparaissent en revanche la nuit, où les programmes peuvent ne pas séparés par des inter-programmes. Un autre problème délicat est que la qualité des résultats liés aux répétitions dépend de l'ensemble de vidéos de référence qui sert à la reconnaissance (EVR). Il est nécessaire que cet EVR soit suffisament complet pour que la segmentation soit effectivement de bonne qualité. Il y a donc nécessité de mettre à jour cet EVR au fur et à mesure, soit de manière manuelle ou semi-automatique, comme nous le proposerons dans le chapitre 5.

Chapitre 4

Étiquetage de programmes

4.1 Introduction

La segmentation en programmes/inter-programmes décrite dans le chapitre 3 est une première étape importante, car elle permet un découpage du flux en segments pertinents. Toutefois, ces segments ne comportent pour l'instant aucune autre information supplémentaire. Il est alors nécessaire d'attacher une information plus sémantique à ces segments, afin de pouvoir naviguer plus aisément, ou de pouvoir faire des recherches simples. À l'instar des magazines spécialisés de télévision qui fournissent les guides des programmes de télévision, et qui permettent une navigation et un choix manuel des programmes, ces segments peuvent être caractérisés par leur genre (film, jeu...), et leur titre. Bien qu'assez simple, ces informations permettent d'organiser le flux, de sorte qu'il est alors facile de retrouver un programme particulier, ou de naviguer dans la grille des programmes. Ce chapitre est dédié à l'acquisition d'une telle information, afin de produire automatiquement un guide des programmes recalé avec le flux. On pourra se reporter au schéma fonctionnel de la figure 1, page 20, pour voir la place de l'étiquetage dans le processus de structuration global.

Nous parcourons brièvement les méthodes existantes qui permettent d'inférer un genre ou un titre d'un programme de télévision. Quelques travaux, peu nombreux, ont proposé d'inférer automatiquement le genre d'une émission à partir de caractéristiques bas et moyen niveau. Jasinschi et al. [JL01] déterminent le genre d'un segment vidéo à partir d'un descripteur formé à partir des probabilités d'appartenance à différentes classes audio (parole, musique, etc...). Les auteurs de [JBR04] extraient une quinzaine de descripteurs audio et vidéo provenant de MPEG-7, et classent les vidéos en 5 genres en utilisant un arbre de décision. Ils obtiennent de très bons résultats. Taskiran et al. [TPBD03] forment un descripteur de plan en extrayant des caractéristiques assez simples, la longueur du plan, un indicateur de mouvement, un de couleur et un de texture. Une phase d'apprentissage est ensuite réalisée, qui détermine un certain nombre de classes par clustering, en ayant préalablement modélisé les descripteurs par un modèle de mélange de gaussiennes. Un modèle hybride modèle de Markov caché et grammaire hors-contexte stochastique est finalement utilisé pour la classification de vidéo en 4

genres, à partir des clusters précédemment trouvés. Là encore, les résultats sont excellents.

Ces méthodes ont l'avantage de pouvoir caractériser assez finement un programme, en le catégorisant éventuellement en sous-classes. Elles sont, en revanche, assez complexes, et il semble délicat de les appliquer sur de très grandes masses de données, et avec un beaucoup plus grand nombre de classes.

Nous proposons d'utiliser un système beaucoup plus simple, qui utilise l'information provenant du guide des programmes. Le guide comporte toujours le titre du programme, parfois le genre, et éventuellement des informations complémentaires, généralement pour des programmes conséquents. Citons, par exemple, les acteurs, le(s) présentateur(s), un résumé, une critique... Ces informations peuvent être récupérées automatiquement par une édition électronique du guide de programme classique, via le site internet de la chaîne par exemple, ou soit directement à partir du guide de programme électronique, diffusé en même temps que le flux vidéo. On utilise, en général, l'acronyme anglo-saxon EPG, Electronic Program Guide, pour faire référence à ce type de guide. Les informations contenues dans l'EPG sont, malheureusement, généralement plus pauvres que dans les guides des magazines. Par exemple, dans DVB [DVB03], qui est actuellement le standard utilisé en télévision numérique, l'information concernant les programmes est transmise dans une table de signalisation, plus précisément la table EIT, Event Information Table, qui décrit les programmes courants et à venir. Les informations concernant le programme sont assez pauvres et se résument au genre et au titre du programme. Les futures normes de description telles TV-anytime [TVA02] permettent de décrire de façon à la fois simple et beaucoup plus détaillée un ensemble de programmes de télévision (acteurs, réalisateur, restrictions, rediffusions...). Des informations plus détaillées seront donc peut être disponible au fur et à mesure de l'adoption de la technologie.

La précision des informations dépend cependant surtout des informations fournies par les chaînes et par les diffuseurs. Ces informations se limitent généralement au titre, et parfois au genre, sur le modèle du guide des programmes papier. Nous n'utiliserons ici que l'information de titre pour étiqueter le flux, mais tout autre type d'information peut être ajoutée comme complément, si disponible. En ce qui concerne le choix d'un guide de type prévisionnel, ou de type instantané (voir introduction, page 19), nous choisissons de travailler avec un guide de type prévisionnel, parce que plus répandu, et que c'est le cas le plus difficile. L'approche que nous proposons reste valide avec un guide de type instantané, avec, on peut l'espérer, de meilleurs résultats.

L'approche classification habituellement proposée [TPBD03, JBR04, JL01] peut éventuellement être utilisée en complément d'une méthode basée sur le guide, à des fins de vérification, ou afin d'obtenir une caractérisation plus fine du programme.

Toutefois, l'utilisation des informations du guide n'est pas immédiate, en raison du manque de précision des horaires indiqués : des retards de plusieurs dizaines de minutes peuvent apparaitre, certains programmes sont manquants ou erronés. L'imprécision du guide des programmes est montrée dans l'annexe F. Il existe une catégorie à part entière de programmes de courte durée, qui ne sont généralement pas présent dans le guide et qui font parfois office de « fusible » pour les chaînes. Ces programmes permettent de réguler l'avance et le retard dans la diffusion en les supprimant, ou en les ajoutant.

Ces programmes sont parfois appelés *programmes interstitiels*, et c'est la dénomination que nous utiliserons. Afin de gérer les erreurs, les imprécisions horaires du guide de programme, et l'absence de ces programmes interstitiels, il est nécessaire de développer une méthode qui puisse *aligner*, ou *recaler*, les informations du guide et la segmentation en programmes. C'est l'objet de ce chapitre.

4.2 Méthode d'utilisation du guide des programmes

Cette section propose une méthode pour aligner le guide des programmes sur la segmentation en programmes, réalisée par la technique exposée au chapitre 3. Il s'agit d'affecter à chaque segment l'étiquette du guide qui lui convient.

4.2.1 Alignement par Dynamic Time Warping

Une technique intéressante pour aligner des séquences tout en intégrant des règles a priori est la distance d'édition[1] utilisée classiquement pour aligner des chaînes de caractères [Lev65]. La distance d'édition est définie comme le coût minimal de la suite de transformations à appliquer pour passer d'une séquence à une autre. Les transformations généralement définies sont l'insertion, la suppression et la substitution, mais d'autres types de transformations peuvent être définies, en fonction de l'application. Les coûts des différentes transformations sont aussi à définir en fonction de l'application. L'intérêt majeur de la distance d'édition et de ses dérivés est de réaliser un alignement global entre les deux séquences, c'est à dire qu'elle permet d'identifier le chemin de coût minimal qui permet de transformer une séquence en une autre, tout en ayant un moyen simple de définir ce critère de coût minimal, à travers les fonctions de coûts.

Nous nous limitons ici aux trois types de transformations standard, l'insertion, la substitution, la suppression, et nous présentons l'algorithme sous la forme donnée par Sakoe et Chiba [SC78], et généralement appelée *Dynamic Time Warping* (DTW) ou, en français, *déformation dynamique temporelle.*

Pour deux séquences $X = (x_0 \dots x_N)$ et $P = (p_0 \dots p_M)$, calculer la DTW revient alors à déterminer la suite de transformations de coût minimal qui permet de passer d'une séquence à une autre. Pour déterminer ce coût minimal, on remplit une matrice des distances :

$$D(i,j) = DTW(X_i, P_j) \quad \text{où } X_i = (x_0 \dots x_i)\ ,\ P_j = (p_0 \dots p_j)$$

Cette matrice est calculée de proche en proche en faisant appel à une méthode de programmation dynamique, qui consiste à dire que le coût en (i,j) ne dépend que des coûts de l'étape précédente :

$$D(i,j) = \min \begin{cases} D(i-1,j-1) + c_{sub}(x_i, p_j) \\ D(i,j-1) + c_{sup}(x_i, p_j) \\ D(i-1,j) + c_{ins}(x_i, p_j) \end{cases}$$

[1] appelée aussi distance de Levenshtein.

où $c_{sub}, c_{sup}, c_{ins}$ sont respectivement les coûts de substitution, de suppression et d'insertion. Ces coûts peuvent éventuellement dépendre du temps. L'utilisation de la programmation dynamique permet de se ramener à un algorithme en $O(mn)$, alors qu'une application brutale par la formule de récurrence produit un algorithme de complexité exponentielle. Lorsque la matrice est entièrement remplie, la distance entre les deux séquences X et P est alors simplement donnée par la valeur de $D(N, M)$.

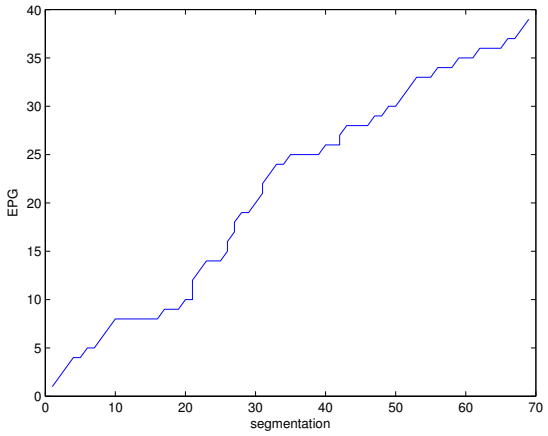

FIG. 4.1 – Exemple de chemin déterminé par DTW entre une segmentation et le guide des programmes. Réalisé ici sur la journée entière du 16/05/2005.

Dans le cas où les séquences sont de même taille, la DTW peut être considérée comme une généralisation de la distance de Hamming. On peut montrer en particulier que la distance de Hamming en est un majorant, $DTW(X, P) \leq D_H(X, P)$.

Si l'on souhaite déterminer le chemin qui a permis de générer cette distance, il faut construire, en parallèle de la matrice des coûts, une matrice des chemins qui indique quelle a été la transformation choisie à chaque instant. Une case de la matrice des chemins contient un symbole indiquant la direction choisie lors du calcul des coûts. Il y a donc autant de de symboles que de transformations possibles. Ici, il y a donc trois symboles de direction possibles : G (gauche), B (bas), D (diagonal), et un symbole d'interdiction $\otimes$, qui sert à indiquer que la case est interdite. Ce dernier symbole n'est pas utilisé dans la version classique de l'algorithme, nous verrons la raison de son introduction à la section 4.3.1. Pour déterminer le chemin, il suffit ensuite de parcourir la matrice des chemins à l'envers, c'est à dire du point (N, M) au point $(1, 1)$, en suivant les indications des symboles. Un exemple de chemin est donné sur la figure 4.1. Ce chemin est un exemple réel d'alignement par DTW entre une segmentation et un guide des programmes, sur la journée du 16/05/2005.

4.2.2 Application de la DTW à l'étiquetage

Nous définissons une segmentation en programmes $X_i = \{x_0 \dots x_i\}$ et le guide de programme associé par $P_j = \{p_0 \dots p_j\}$. Chaque élément de X_i et de P_i est un couple de valeur indiquant le début et la fin du programme $x_i = (x_i^d, x_i^f)$. L'application de la DTW à notre problème d'alignement est alors presque immédiate. Il suffit de définir les coûts pour les différentes transformations.

Les fonctions de coût $c_{sub}, c_{sup}, c_{ins}$ sont à définir et sont donc l'endroit idéal où intégrer des informations a priori du domaine. Un exemple d'information a priori est que la structure des émissions la nuit est complètement différente (peu ou pas de publicités, retard important, suppression de programmes, etc...) et cela peut se traduire dans les fonctions de coût. En fonction de la chaîne ou de l'horaire, les coûts peuvent donc éventuellement être modifés, tout en conservant le même algorithme général.

Les fonctions de coût sont définies par l'intermédiaire d'une distance locale d entre composantes des séquences X et P.

$$\begin{aligned} c_{sub}(x_i, p_j) &= \gamma d(x_i, p_j) \\ c_{sup}(x_i, p_j) &= d(x_i, p_j) \\ c_{ins}(x_i, p_j) &= d(x_i, p_j) \end{aligned}$$

Classiquement, $\gamma = 2$. Cependant pour privilégier une substitution par rapport à une suppression puis une insertion, on doit vérifier $c_{sub}(x_i, p_j) < c_{ins}(x_i, p_j) + c_{sup}(x_i, p_j)$ d'où $\gamma < 2$. Nous utilisons en pratique $\gamma = 1,5$.
Nous définissons la distance locale d par :

$$d(x_i, p_j) = \alpha |p_j^f - p_j^d - (x_i^f - x_i^d)| + \beta \left[|p_j^d - x_i^d| + |p_j^f - x_i^f| \right]$$

Cette distance est composée de 2 termes. le premier, $|p_j^f - p_j^d - (x_i^f - x_i^d)|$, mesure la similarité de la longueur des segments x_i et p_j, tandis que le deuxième terme, $|p_j^d - x_i^d| + |p_j^f - x_i^f|$, mesure la similarité des horaires de diffusion. La seule information disponible pour effectuer l'alignement est, en effet, l'horaire de début et de fin du programme. La distance locale, et donc indirectement, les coûts sont donc proportionnels à la proximité temporelle du programme et du segment, ainsi qu'à leur similarité de longueur. La figure 4.2 donne un exemple d'alignement par DTW, avec $\alpha = \beta = 1$.

Une optimisation possible serait d'apprendre les valeur de α et β qui produisent les meilleurs résultats d'étiquetage, en différenciant ces deux paramètres pour les trois fonctions de coûts. C'est une optimisation utilisée dans diverses applications [CA04, SB04], et basée sur les travaux de Ristad et al. [RY98] sur l'apprentissage de distance d'édition. Nous n'avons pas cherché à développer ce genre de méthode, pour deux raisons. La première est que quelques tests nous ont convaincus de son influence marginale sur les résultats. L'importance de la proximité temporelle est, en effet, à peu près la même que celle de la similarité de longueur. Il y a de plus un risque de biais des coûts par rapport au corpus. Ce biais peut être bénéfique si l'apprentissage est réalisé pour une chaîne donnée, mais on perd alors en généricité. La deuxième raison est que nous pensons que l'amélioration des résultats d'étiquetage passe essentiellement par l'intégration

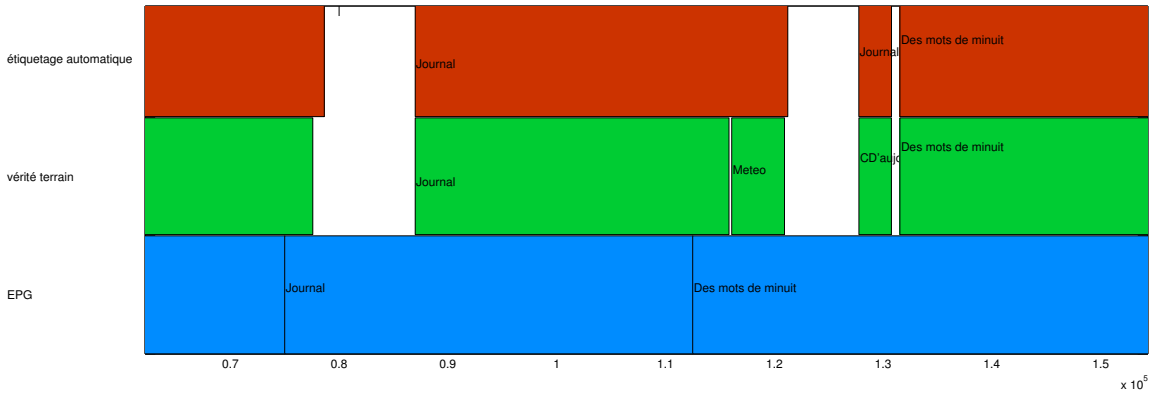

FIG. 4.2 – Exemple d'étiquetage par DTW.

d'informations extérieures et, en particulier, l'utilisation de l'information de répétition de certains segments. L'utilisation de cette information de répétition est l'objet de la section 4.3. Compte tenu de ces remarques, les coefficients de proportionnalité sont choisis uniformes, $\alpha = \beta = 1$.

La section suivante propose quelques pré-traitement possibles sur le guide de programmes, qui peuvent éventuellement améliorer les résultats.

4.2.3 Pré-traitement du guide des programmes

Afin d'effectuer l'alignement par DTW, il peut être profitable de pré-traiter le guide des programmes, afin de le rendre plus propice à un alignement. Une règle simple est de diviser les programmes supérieurs à une certaine durée. Expérimentalement, il est observé que les programmes longs sont divisés en plusieurs parties, afin d'insérer de la publicité. Cette division des programmes en 2, 3 voire même jusqu'en 10 parties, est très préjudiciable pour l'alignement, puisqu'il est évidemment difficile pour la DTW de trouver une correspondance entre 10 programmes courts et un programme très long. La règle utilisée, en pratique, est de diviser les programmes de plus d'une 1h15 en deux parties, et les programmes de plus de deux heures en trois parties. Cette heuristique est appelée **Division**.

A l'inverse, les programmes courts, indiqués dans le guide avec une durée très peu précise (la précision du guide est de 5 minutes), peuvent bruiter l'alignement, les étiquettes de ces programmes ayant tendance à se répartir sur les programmes adjacents. L'heuristique appellée **Suppression** consiste à supprimer les programmes du guide d'une durée inférieure ou égale à 5 minutes.

C'est aussi l'endroit éventuel pour utiliser des a priori sur la chaine. Par exemple, les chaines publiques françaises ont l'interdiction de couper les films et téléfilms en plusieurs parties. Cette connaissance peut donc être mise à profit pour produire un guide plus proche de la réalité. Sur les chaines privées, les films seront découpés en plusieurs morceaux, mais non sur les chaines publiques. Sachant que l'on sait quelle chaine on

traite, cette information a priori est d'une utilisation raisonnable. Cette heuristique est appelée **ChainePublique**, elle n'a évidemment de sens que lorsqu'elle est utilisée en combinaison avec l'heuristique **Division**.

Les performances de ces différentes heuristiques sont données sur la figure 4.3. Ces courbes donnent les résultats de l'étiquetage sur l'ensemble du corpus 2. Nous donnons ces résultats à titre indicatif, et afin de justifier le choix d'une des méthodes de prétraitement, sur lesquelles nous ne reviendrons pas. Les mesures utilisées pour le calcul des résultats sont définies à la section 4.6.2.

De nombreux autres traitements peuvent être imaginés, mais ces heuristiques ne sont pas des solutions satisfaisantes. Une meilleure solution consiste à construire un guide prédictif, à partir du guide prévisionnel, non pas avec des heuristiques a priori, mais en réalisant un apprentissage de la grille recalée d'une chaine. La méthode proposée par Poli [Pol07] s'insère parfaitement dans cette optique. Cette solution n'a pas pu être testée, faute de disponibilité des données, mais il est très probable que l'amélioration avec un tel guide prédit soit non-négligeable.

A défaut d'avoir réalisé une telle prédiction du guide, nous nous contentons d'un minimum de règles simples. Nous n'utilisons, en pratique, que l'heuristique **Division**, qui est la plus naturelle, et parmi celle qui donne les meilleurs résultats.

La section suivante présente comment améliorer l'étiquetage en intégrant une information extérieure dans le mécanisme de la DTW.

4.3 Intégration d'informations de reconnaissance

4.3.1 Présentation de la DTW ancrée

Un des moyens de s'affranchir du peu de précision de la macro-segmentation, et de corriger les erreurs de l'alignement, est d'utiliser une méthode de reconnaissance de séquences vidéo, de façon à détecter les répétitions. Les séquences répétées sont, en général, soit des inter-programmes ou soit des génériques d'émissions, et sont donc utiles pour identifier les instants d'inter-programmes, ainsi que les débuts et fins d'émissions. Nous utilisons la méthode présentée au chapitre 2 pour reconnaitre ces segments répétés.

Ces informations de reconnaissance sont clairement utiles à l'étiquetage du flux. Le problème est que la manière d'intégrer ces informations n'est pas évidente. Utiliser les reconnaissances comme post-traitement pour confirmer ou infirmer l'alignement effectué par DTW est toujours possible, mais n'est pas vraiment satisfaisant : il n'est pas évident de fusionner les résultats, et cette fusion perdrait certainement le bénéfice d'un alignement global en se rapprochant plus d'une méthode de décision segment par segment. Il serait préférable que les reconnaissances puissent guider le processus d'alignement. Nous proposons une telle méthode dans cette section.

La DTW se prête bien à cette intégration d'information. Il suffit de considérer que la reconnaissance revient à étiqueter quelques parties du flux de façon certaine. Ceci se traduit dans l'algorithme d'alignement par des points de passage obligatoires. Ces points, appelés **ancres**, permettent aussi de contraindre l'espace de recherche sans avoir

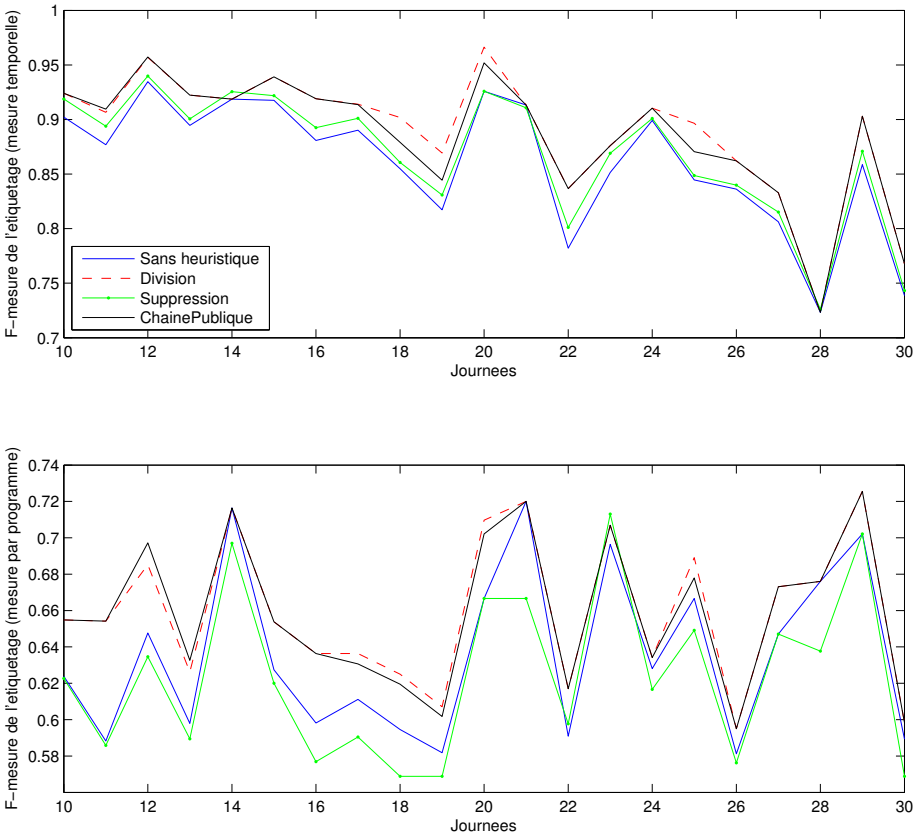

FIG. 4.3 – Performances comparées des différentes heuristiques de pré-traitement du guide des programmes.

recours à une restriction arbitraire de l'espace, comme utilisé parfois avec une DTW [SC78]. Le processus total est alors appelé **DTW ancrée**.

Supposons qu'un plan ait été reconnu et que ce plan soit localisé dans le segment x_i de la segmentation en programmes X. Ce plan possède une étiquette, qui est alors cherchée dans le guide des programmes. Si cette étiquette est présente dans le guide, dans un voisinage autour de la position du plan reconnu, on obtient un programme p_j, sinon l'ancrage n'est pas possible[2], et on se ramène alors à une DTW classique. À l'aide de la position de ce programme p_j, et du segment x_i, nous obtenons alors une ancre au point (i, j), par laquelle nous souhaitons contraindre le chemin à passer. La

[2]on peut choisir d'ajouter artificiellement un programme p_j à l'EPG avec l'étiquette du segment reconnu. Cette solution est délicate à mettre en oeuvre, car on ne sait pas si l'on doit ajouter un programme, ou remplacer l'étiquette existante. Cette solution a été abandonnée au profit de la méthode 4.4

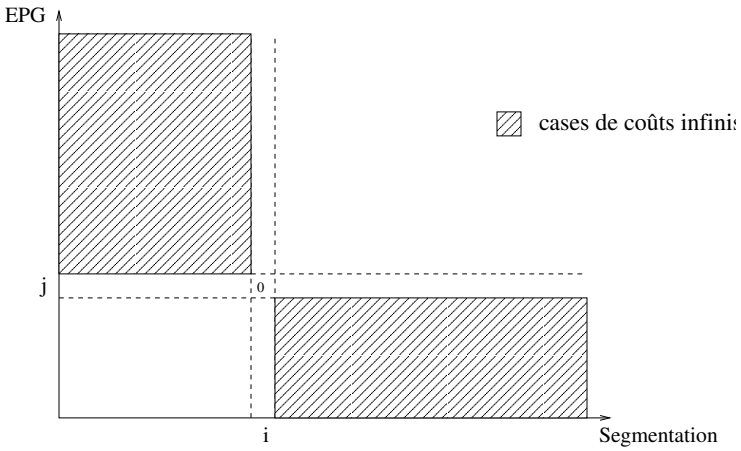

FIG. 4.4 – Présence d'une ancre au point (i, j) dans la matrice des coûts de la DTW avec une politique souple.

détermination de cette position se fait, nous l'avons dit, dans un voisinage autour de la position du plan reconnu. Il est nécessaire de faire une recherche dans un certain voisinage, car le guide étant erroné, l'annonce du programme peut être assez éloignée de la position de la reconnaissance. La détermination de la position de l'ancre ne peut pas se faire non plus sur tout le guide, car certains programmes sont diffusés plusieurs fois pendant la journée, on risquerait alors d'aligner un programme avec une de ses rediffusions, ce qui serait catastrophique pour l'ensemble de l'étiquetage. En pratique, le voisinage est centré autour du plan reconnu, d'une durée de 2 heures, une heure avant la reconnaissance et une heure après. La durée de ce voisinage n'est pas une valeur critique : les possiblités de confusion sont assez rares.

Nous considérons deux politiques d'ancrage : une politique d'ancrage souple et une politique d'ancrage restrictive. L'ancrage souple est réalisé en pré-remplissant la matrice des coûts par :

$$\begin{cases} d(x_i, p_j) = 0 \\ d(x_l, p_k) = \infty \quad \forall (k,l) \text{ tels que } (k < j \text{ et } l > i) \text{ ou } (k > j \text{ et } l < i) \end{cases}$$

La figure 4.4 illustre la politique d'ancrage souple, avec une ancre au point (i, j). On peut voir sur cette figure que cette politique de pré-remplissage de la matrice laisse la possibilité de ne pas passer par le point (i, j). Il y a cependant de grandes chances pour que le chemin final passe par le point d'ancrage puisqu'il est de coût nul. Cette politique d'ancrage est appelée souple car elle laisse libre les chemins autour de l'ancre, et y compris donc, la possibilité de ne pas passer par l'ancre elle-même.

L'ancrage restrictif est défini par :

$$\begin{cases} d(x_i, p_j) = 0 \\ d(x_l, p_k) = \infty \quad \forall (k,l) \text{ tels que } (k < j \text{ et } l \geq i) \text{ ou } (k > j \text{ et } l \leq i) \end{cases}$$

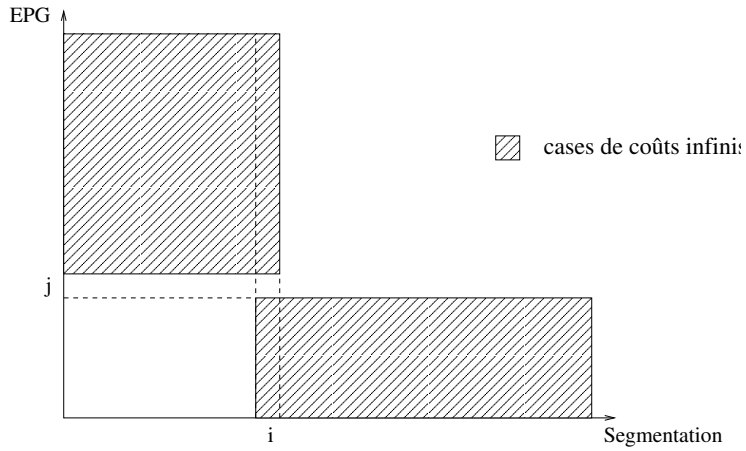

FIG. 4.5 – Présence d'une ancre au point (i, j) dans la matrice des coûts de la DTW avec une politique restrictive.

La figure 4.5 illustre cette politique d'ancrage. Il n'y a ici plus de passage possible ailleurs que par le point d'ancrage. Cette politique est évidemment beaucoup plus restrictive, il y a moins de degré de liberté autour du point d'ancrage, ce qui peut être pénalisant.

Indépendamment de la politique d'ancrage choisie, et parallèlement au pré-remplissage de la matrice des coûts, la matrice des chemins est elle aussi pré-remplie. Les cases correspondantes aux infinis de la matrice des coûts sont remplies par des symboles d'interdiction dans la matrice de chemins, indiquant que le chemin ne doit pas passer par cette case. L'étape de pré-remplissage de la matrice est résumée par l'algorithme 4.

Suite à ce pré-remplissage, la matrice des coûts est ensuite calculée de manière classique, par la méthode exposée en 4.2.2. La seule différence est que seules les cases ne comportant pas de signe infini ou de zéro sont calculées, et de même pour la matrice des chemins : les cases gratuites et les cases interdites ne sont pas considérées.

4.3.2 Problèmes de recouvrement

Dans la politique d'ancrage restrictive, des problèmes peuvent apparaitre lorsque des reconnaissances sont trop proches les unes des autres. Il peut alors apparaitre un phénomène de recouvrement, illustré par les figures 4.6 et 4.7.

4.3.2.1 Recouvrement de type 1

Un premier recouvrement, appelé recouvrement de type 1, apparait lorsque deux reconnaissances appartiennent au même segment, comme montré sur la figure 4.6. Ceci arrive, par exemple, lorsque la segmentation est erronée, et qu'un segment couvre en fait deux programmes. Il peut alors exister deux reconnaissances à l'intérieur de ce segment, avec des étiquettes différentes. Il n'existe alors plus de « chemin libre », puisque

Fonction init_et_place_ancres() :

D : **Matrice** N**x**M ; *[matrice des coûts]*
C : **Matrice** N**x**M ; *[matrice des chemins]*
$\otimes$: **Symbole d'interdiction** ;
Pour $i = 1$ à N **faire** $D(i,1) = c_{ins}(i,1)$ **Fin Pour**
Pour $j = 1$ à M **faire** $D(1,j) = c_{sup}(1,j)$ **Fin Pour**
$D(1,1) = 0$
Pour chaque ancre en position (i,j) **faire**
  $D(i,j) = 0$
  Pour k **de** 1 **à** $j-1$ **faire**
    Pour l **de** $i+1$ **à** M **faire**
      $D(k,l) = \infty$
      $C(k,l) = \otimes$
  Pour k **de** j+1 **à** N **faire**
    Pour l **de** 1 **à** $i-1$ **faire**
      $D(k,l) = \infty$
      $C(k,l) = \otimes$
Fin Pour

Fin

Algorithme 4: Initialisation de la matrice des coûts et placement des ancres, politique restrictive.

la colonne i est entièrement constituée d'infinis. Le comportement de l'algorithme est alors imprédictible.

Une solution simple serait de mettre à zéro la case (i,j) ou la case $(i,j+1)$, mais le choix de l'une ou l'autre des cases (et donc des reconnaissances) est alors arbitraire. Nous adoptons une solution un peu plus complexe, qui consiste, en lieu et place des coûts infinis, de calculer effectivement les coûts par l'algorithme classique de la DTW, mais en les pénalisant fortement par une valeur considérée comme inatteignable, remplissant un rôle de valeur infinie. Les cases de la matrice des chemins sont par contre toujours remplies de la manière expliquée en 4.3.1, c'est à dire qu'elles sont remplies par des symboles interdits. Le meilleur chemin est alors déterminé normalement, en parcourant la matrice des chemins à l'envers, de la case (N,M) à la case $(0,0)$. La différence par rapport au cas présenté en 4.3.1, est que si un recouvrement de type 1 est présent au point (i,j), la détermination du chemin va générer un symbole interdit lors du passage dans la colonne i. Le chemin est donc bloqué. La résolution de ce problème consiste à déterminer a posteriori la prochaine meilleure position, en la déterminant à partir de la

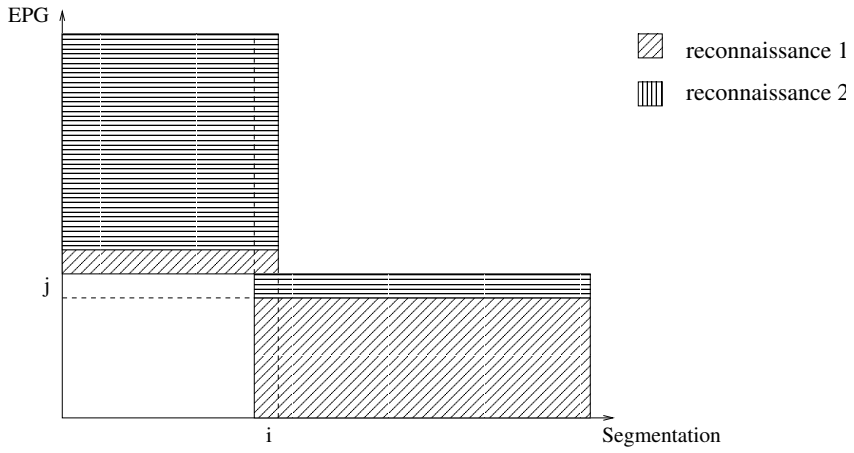

FIG. 4.6 – Problème de recouvrement de type 1

matrice des coûts par :

$$(k,l) = \arg\min\left(D(i-1,j-1), D(i,j-1), D(i-1,j)\right)$$

Cette détermination est possible grâce au fait que, comme précemment expliqué, les coûts des cases interdites ne sont pas uniformes, ni infinis, mais calculées par DTW avec une pénalisation. Cette détermination du chemin, position par position, se poursuit tant que l'on ne retombe pas sur une case non-interdite.

L'ensemble de la méthode est résumé par l'algorithme 4 pour l'initialisation et le placement des ancres, l'algorithme 5 pour le calcul de la matrice des coûts, en intégrant la méthode de résolution des recouvrements de type 1. La détermination du chemin est donnée par l'algorithme 6.

4.3.2.2 Recouvrement de type 2

Un recouvrement de type 2 apparait lorsque deux ancrages sont successifs dans la segmentation, mais non successifs dans le guide. Ceci est illustré sur la figure 4.7. La conséquence est aussi qu'il n'existe plus de chemin libre dans la matrice des coûts. On peut choisir d'appliquer la même solution que pour les recouvrements de type 1. Néanmoins, dans ce cas, ce ne sont pas les points d'ancrages eux-mêmes qui sont recouverts, le choix est donc moins crucial. Nous choissions ici simplement de mettre à zéro la case $(i+1,j)$[3] ce qui revient à une suppression du segment j de l'EPG, et permet à un chemin d'exister.

[3]Nous aurions pu faire exactement la même chose avec la case $(i, j+1)$, qui permet aussi d'ouvrir un chemin. Le choix est ici arbitraire.

Fonction Remplir_matrice_coûts() :

D : **Matrice** NxM ;
Pour i **de** 2 **à** N **faire**
Pour j **de** 2 **à** M **faire**
Si $(D(i,j) = \infty)$ **Alors**
$D(i,j) = \min\left(c_{sub}(i,j) + \infty, c_{ins}(i,j) + \infty, c_{sup}(i,j) + \infty\right)$
Sinon
Si $(D(i,j) \neq 0)$ **Alors**
$$D(i,j) = \min \begin{cases} D(i-1,j-1) + c_{sub}(i,j) \\ D(i,j-1) + c_{sup}(i,j) \\ D(i-1,j) + c_{ins}(i,j) \end{cases}$$
Fin Si
Fin Si
Fin Pour
Fin Pour

Fin

Algorithme 5: Calcul de la matrice des coûts par la DTW ancrée, modifiée pour prendre en compte les recouvrements de type 1

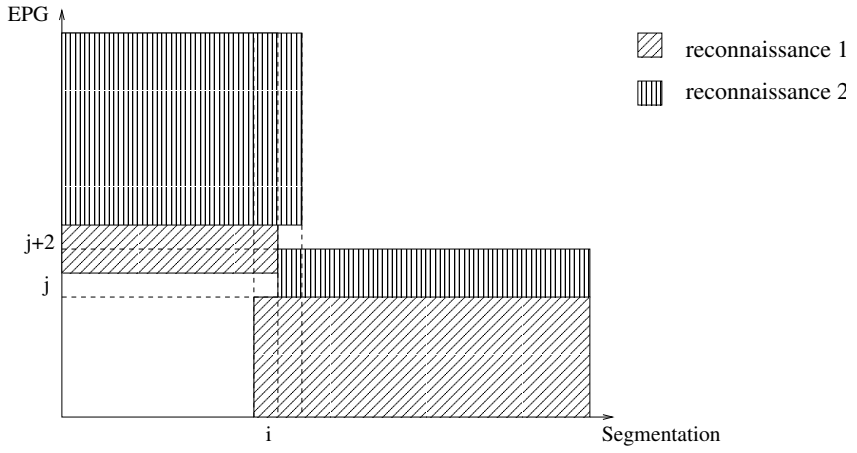

FIG. 4.7 – Problème de recouvrement de type 2

4.4 Résolution d'ambiguités

La DTW ancrée ne peut résoudre toutes les inexactitudes de l'étiquetage. Sa principale restriction est que l'étiquette du programme à aligner doit être présente dans le guide des programmes, ce qui n'est pratiquement jamais le cas pour les programmes interstitiels, qui sont aussi les plus difficiles à étiqueter. La DTW ancrée ne peut donc

```
Fonction Détermination_chemin() :
    C : Matrice NxM ; [matrice des chemins]
    D : Matrice NxM ; [matrice des coûts]
    Pour i de N à 1 faire
        Pour j de M à 1 faire
            Selon que
                C(i,j) = G [gauche]
                    i=i-1 ;
                C(i,j) = B [bas]
                    j=j-1 ;
                C(i,j) = D [diagonal]
                    i=i-1 ; j=j-1 ;
                C(i,j) = ⊗ [interdit]
                    (i,j) = arg min (D(i-1,j-1), D(i,j-1), D(i-1,j))
            Fin Selon que
        Fin Pour
    Fin Pour
Fin
```

Algorithme 6: Détermination du chemin à partir de la matrice des chemins, avec la méthode de résolution des recouvrements de type 1.

pas traiter le cas où une reconnaissance est présente dans un segment de la segmentation automatique, mais sans qu'il n'y ait un segment d'étiquette correspondante dans le guide. Nous proposons, dans cette section, une méthode pour utiliser malgré tout l'information de reconnaissance dans ce cas.

Cette méthode se déroule après l'application de la DTW ancrée. C'est donc une méthode de post-traitement, qui cherche à résoudre les contradictions éventuelles entre l'étiquetage proposé par la DTW ancrée, et les étiquettes provenant de la reconnaissance. Nous appelons cette méthode « résolution d'ambiguités », car elle permet de résoudre les ambiguités sur certains segments, c'est à dire qu'elle étudie pour chaque segment étiqueté par la DTW ancrée, si son étiquette est conforme aux étiquettes des reconnaissances éventuellement présentes dans ce segment.

Cette méthode est particulièrement utile lorsque l'étiquette correcte n'est pas présente dans le guide des programmes, ce qui est fréquent. L'étiquetage effectué par la DTW ancrée est alors évidemment faux. Il peut cependant arriver pour certains types de programmes, par exemple les films ou téléfilms, que la DTW ancrée produise un étiquetage correct et que la reconnaissance soit erronée. Afin de prendre une décision, nous proposons d'utiliser un critère probabiliste, basé sur des a priori liés aux types d'erreurs de la reconnaissance.

Nous exposons à présent la méthode. Nous définissons deux hypothèses :
- H_0, l'étiquette correcte provient de la reconnaissance.
- H_1, l'étiquette correcte provient de la DTW ancrée.

Étant donné une observation O, la décision est prise via un test d'hypothèse bayésien :

$$\frac{P(O|H_1)}{P(O|H_0)} > \frac{P_0}{P_1} \text{ alors } H_1 \text{ sinon } H_0$$

où P_i la probabilité à priori de l'hypothèse H_i.

Afin d'estimer $P(O|H_i)$, l'observation O est considérée constituée d'observation élémentaires indépendantes o_k. Ces observations élémentaires sont ensuites faciles à estimer en utilisant un corpus d'apprentissage. On a alors :

$$P(O|H_i) = \prod_k p(o_k|H_i)$$

Trois observations élémentaires sont définies o_1, o_2 et o_3.
- o_1 est la longueur du segment et sa distribution de probabilité est considérée gaussienne. L'intérêt de cette observation provient du fait que les reconnaissances erronées sont généralement situées sur des longs segments, alors que les reconnaissances situées sur des petits segments sont généralement justes.
- o_2 et o_3 sont des observations binaires, vraies lorsqu'une reconnaissance est située respectivement au début, et à la fin du segment. o_2 et o_3 sont définies pour prendre en compte le fait que les génériques de début et de fin sont souvent bien détectés, et donc que la présence d'une reconnaissance en début ou fin de segment est un bon indicateur que l'étiquette de la reconnaissance est exacte.

Les probabilités conditionnelles des observations élémentaires sont estimées par apprentissage sur la journée du 9/05/2005, qui sert aussi d'EVR. Les paramètres des deux lois gaussiennes $p(o_1|H_0)$ et $p(o_1|H_1)$ sont estimées de manière classique par estimation du maximum de vraisemblance (voir annexe C). Les probabilités conditionnelles de o_2 et o_3 sont estimées directement par comptage, de même pour les probabilités a priori.

Pour résumer, le critère de décision est donc :

$$\frac{\prod_{k=1}^{3} p(o_k|H_1)}{\prod_{k=1}^{3} p(o_k|H_0)} > \frac{P_0}{P_1} \text{ alors } H_1 \text{ sinon } H_0$$

Dans le cas où la reconnaissance produit plusieurs étiquettes différentes appartenant à un même segment, et si une des étiquettes est majoritaire, c'est à dire qu'elle apparait plus de fois que les autres, alors c'est elle qui est choisie, sinon le choix est arbitraire.

Les résultats de la résolution d'ambiguité sont donnés en section 4.6.5.

4.5 Illustration d'un alignement partiel segmentation-EPG

L'apport de la DTW ancrée est montrée sur la figure 4.8. La figure montre la même portion de segment vidéo que la figure 4.2, où l'alignement était alors effectué par une DTW simple. La différence se situe au niveau du programme « CD'aujourd'hui ».

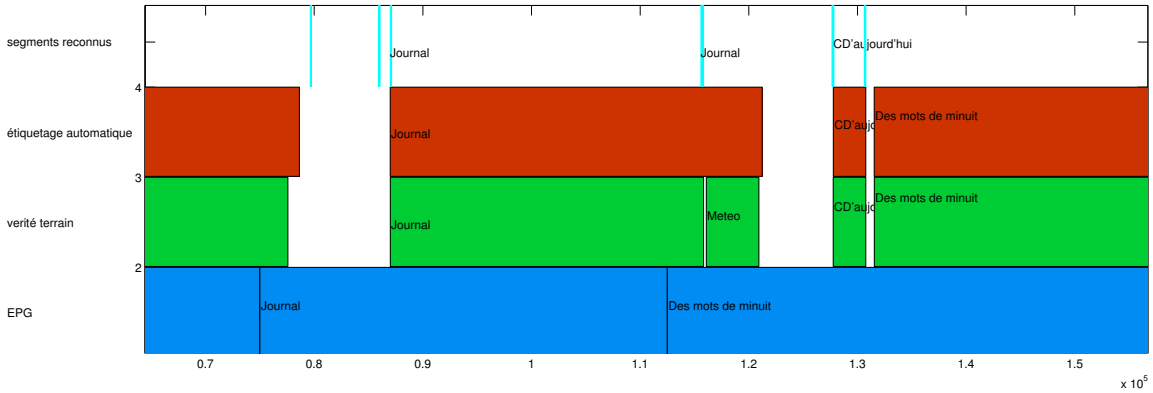

FIG. 4.8 – Alignement par DTW ancrée et résolution d'ambiguités

Cette étiquette n'existe pas dans le guide des programmes, et provient de l'EVR, via une reconnaissance. Ce segment est donc correctement étiqueté grâce à la résolution d'ambiguités.

Notons aussi que, dans ce cas, les deux reconnaissances « journal », servent d'ancres. Leur présence dans le deuxième segment de la segmentation automatique contraint la DTW ancrée à aligner ce segment avec le deuxième segment du guide, le segment « journal », puisqu'il contient la même étiquette que les reconnaissances. L'information de reconnaissance n'est cependant pas encore utilisée à son maximum, puisque le générique de fin du journal a été reconnu, mais ne permet pas de modifier la segmentation, et d'isoler le programme « Météo ».

4.6 Résultats

Cette partie donne les résultats de l'étiquetage du flux sur la totalité du corpus 2, qui est décrit, nous le rappelons, en annexe A. Le protocole expérimental est explicité dans la section 4.6.1, puis les mesures d'évaluation de l'étiquetage sont présentés dans la section 4.6.2. La section 4.6.3 permet de déterminer la meilleure politique d'ancrage, la section 4.6.4 étudie la variation des résultats avec le seuil de classification, c'est à dire l'influence de la segmentation sur les résultats d'étiquetage. La section 4.6.5 évalue l'apport des méthodes proposées de DTW ancrée et de résolution d'ambiguité par rapport à une DTW classique. La section 4.6.6 analyse enfin les résultats de l'étiquetage au cours du temps.

4.6.1 Protocole expérimental

L'alignement est réalisé sur 24 heures de vidéo, soit une journée entière. Le choix de la durée sur laquelle réaliser l'alignement s'est effectué de lui-même, car les guides de programmes sont généralement construits de façon à décrire une journée entière. De plus, les séquences vidéos du corpus 2 sont enregistrées par tranches de 24h, et enfin, 24

heures de télévision produisent environ 70 segments pour 40 programmes, ce qui est bien géré par la DTW. Une longueur de test trop faible réduirait l'intérêt d'un alignement global, et une longueur trop importante en réduirait peut être la précision. L'horaire de début de la journée n'a pas d'importance. À titre d'information, les vidéos du corpus 2 commencent toutes à 15h28.

Le processus d'alignement nécessite trois données d'entrée, toutes représentées sous la forme d'une liste de segments, caractérisés par leurs instants de début et de fin :

- la segmentation résultante du processus décrit au chapitre 3 ;
- le guide des programmes ; les segments sont ici étiquetés ;
- les reconnaissances détectées. Les segments sont ici aussi étiquetés.

Les reconnaissances sont obtenues par la méthode du chapitre 2, en cherchant les plans communs entre la journée à étiqueter et un EVR. Dans l'ensemble des résultats présentés dans ce chapitre, l'EVR est **statique**, c'est à dire que l'EVR ne varie pas. L'EVR utilisé ici est l'ensemble de la journée du 9/05/2005, étiquetée manuellement. Cet EVR n'est donc pas construit spécifiquement pour la tâche de reconnaissance, et de nombreux inter-programmes ne seront donc pas détectés car non présents dans l'EVR. L'impact de la « qualité » de l'EVR sera abordé au chapitre 5.

4.6.2 Définition des méthodes d'évaluation

L'évaluation de la qualité de l'étiquetage n'est pas évidente. Plusieurs types de problèmes se posent. Tout d'abord, les segments sont de tailles très hétérogènes. On pourrait alors considérer comme normal que l'étiquetage erroné d'un segment de très longue durée (un film par exemple) ait plus d'impact qu'un programme très court. D'autre part, et c'est le problème majeur, les segments de la vérité terrain ne correspondent pas exactement avec ceux de la segmentation automatique. Comment gérer alors les débordements, les sur-segmentations, les sous-segmentations ?

Nous avons choisi de donner deux mesures de qualité de l'étiquetage : une mesure temporelle, indiquant image par image la justesse de l'étiquetage, et une mesure par programme, permettant une vision de plus haut niveau, et sûrement plus proche de la qualité perçue par un humain. Nous définissons maintenant les algorithmes pour calculer ces deux mesures.

4.6.2.1 Mesure temporelle

La mesure temporelle mesure la justesse de l'étiquetage image par image. Pour une image donnée, son étiquette dans la vérité terrain est donc comparée à celle dans l'étiquetage automatique. Seul l'étiquetage des programmes est considéré. Le cas des images classées en tant que programmes dans la vérité et l'étiquetage automatique est simple : c'est une bonne détection si les étiquettes sont égales, sinon c'est une fausse détection. Dans le cas où seule l'image de la vérité terrain est de type programme, alors c'est une détection manquée. La réciproque, image de type programme dans l'étiquetage automatique mais non dans la vérité terrain, est considérée comme une fausse détection.

À partir d'une segmentation X et de la vérité terrain V, nous déterminons tout

d'abord les images communes, données par $X \cap V$. Le nombre d'images de $X \cap V$ ayant une même étiquette dans X et dans V donne N_b le nombre d'images correctement étiquetées. Le nombre d'images de $X \cap V$ n'ayant pas la même étiquette dans X et dans V donne N_{f_1}. Nous déterminons alors $Card(V \setminus X)$, le nombre d'images présentes dans V mais pas dans X, ce qui nous donne N_m, le nombre d'images manquées. Nous calculons enfin $Card(X \setminus V)$, le nombre d'images présentes dans X mais pas dans V, ce qui nous donne N_{f_2}. Le nombre d'images incorrectement étiquetées est alors donné par $N_f = N_{f_1} + N_{f_2}$. Nous pouvons alors calculer les classiques scores de précision et de rappel par :

$$\text{précision} = \frac{N_b}{N_b+N_f} \quad \text{rappel} = \frac{N_b}{N_b+N_m} \tag{4.1}$$

La mesure temporelle permet de bien se rendre compte de l'exactitude de l'étiquetage sur l'ensemble du flux, et cela indépendamment du nombre de programmes, ou de leur longueur. Elle donne le pourcentage d'étiquetage correct en terme de durée. Elle a l'avantage de ne pas être perturbée par des segmentations différentes entre vérité terrain et segmentation automatique.

Cette mesure est en revanche un peu biaisée, parce que les programmes très longs sont en général bien étiquetés. Les scores obtenus par la mesure temporelle sont donc toujours relativement élevés, y compris le score du guide des programmes, que l'on ne peut pourtant pas considérer comme un modèle d'exactitude. De plus, elle ne mesure pas forcément la qualité de l'étiquetage perçue par un humain, qui va plutôt raisonner en termes de *programmes*.

4.6.2.2 Mesure par programme

Nous proposons une mesure complémentaire de l'étiquetage, de plus haut niveau, et peut être plus intuitive. L'unité de classification n'est plus l'image mais le programme. Le problème majeur est que les programmes, dans la vérité terrain et dans la segmentation automatique, n'ont pas les mêmes bornes. Appellons X la segmentation automatique et V la vérité terrain. Il faut alors prendre un soin particulier à prendre en compte les cas où plusieurs segments de X sont inclus dans un seul segment de V et inversement. L'algorithme 7 détaille cette méthode de mesure.

Le principe de la méthode consiste à déterminer, pour chaque segment x_i de X, l'ensemble des segments de vérité terrain qui partagent un support commun suffisant[4]. Nous considérons ces segments de vérité terrain comme des hypothèses. La décision sur l'exactitude de l'étiquetage est réalisée a posteriori, en testant si la plus grande de ces hypothèses possède la même étiquette que le segment x_i courant. L'étiquetage est alors considéré comme juste, mais l'ensemble des autres hypothèses sont comptabilisées en tant qu'étiquetage erronés. Ceci est fait pour prendre en compte les sous-segmentation, comme illustré par la figure 4.9 : le segment *On aime trop la vie* s'étale sur plusieurs segments de la vérité terrain. Dans ce cas précis, pour ce segment, l'évaluation retournera des valeurs de $N_b = 1$ et $N_f = 4$.

[4]La longueur de ce support, *support_ min*, n'a qu'une influence marginale sur la détermination de ces segments.

```
Fonction Mesure_etiquetage_par_programme() :
    X : segmentation ;
    V : vérité terrain ;
    N_b = 0 : entier ;
    N_f = 0 : entier ;
    support_min = 300 : entier ;
    L = ∅; : Liste
    Pour chaque x_i de X faire
        Pour chaque v_j de V faire
            l_j = card(x_i ∩ v_j)
            Si (l_j > support_min) Alors
                L ← l_j
            Fin Si
        Fin Pour
        détermine v_k tel que l_k = max L
        Si (étiquette(v_k) = étiquette(x_i)) Alors
            N_b = N_b + 1;
            N_f = N_f + Card(L) − 1;
        Sinon
            N_f = N_f + 1
        Fin Si
        L = ∅
    Fin Pour
Fin
```

Algorithme 7: Définition de la mesure par programme de l'étiquetage.

À noter que la valeur de N_m, le nombre de segments manqués, n'est pas donnée par cet algorithme. N_m désigne ici les programmes pour lesquels il n'existe pas de segments de X qui correspondent. Pour calculer cette valeur, il suffit de parcourir la vérité terrain et de déterminer les segments de la vérité terrain qui sont d'intersection nulle avec X. Ceci est sans difficulté, et peut se faire, par exemple, lors du calcul de la mesure temporelle.

À partir de N_b, N_f et N_m, la précision et le rappel sont alors calculés par la formule classique 4.1.

4.6.3 Choix de la politique d'ancrage

Nous déterminons dans cette section la meilleure des deux politiques d'ancrage proposées en section 4.3.1.

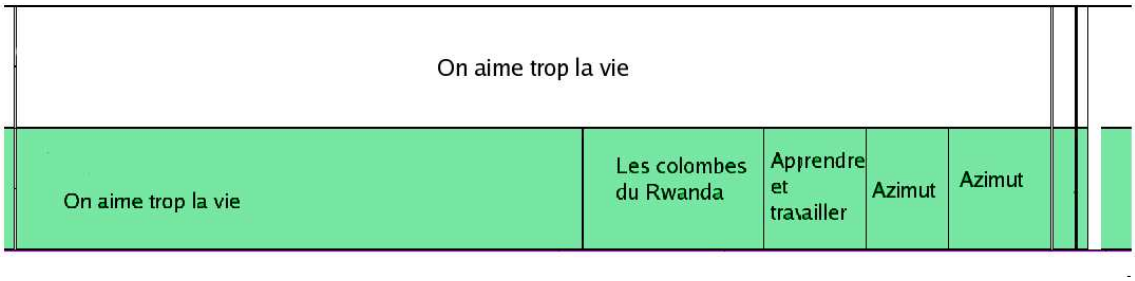

FIG. 4.9 – Exemple de sous-segmentation pour le calcul de la mesure par programme. La vérité terrain est en bas, en vert, et la segmentation et étiquetage automatique est en haut. L'axe des abscisses représente le temps.

La politique souple a l'avantage de ne pas souffrir de problèmes de recouvrement, mais nous avons vu que ces problèmes pouvaient être résolus. Il n'est alors pas évident de choisir quelle politique possède le moins de risques : une politique souple, qui laisse un certain degré de liberté à la DTW pour choisir le meilleur chemin, au risque de ne pas passer par le point d'ancrage, ou une politique plus restrictive, qui assure de passer par le point d'ancrage, mais interdit certains alignements.

Nous tranchons cette question en donnant les résultats d'étiquetage, moyennés sur 20 jours de vidéos, sur la figure 4.10, ainsi que jour par jour, sur la figure 4.11. Nous donnons à chaque fois les résultats d'étiquetage avec la mesure temporelle et la mesure par programme. Les courbes précision en fonction du rappel de la figure 4.10 sont obtenues en faisant varier le seuil de classification, défini en 3.4.1. L'analyse de ces courbes en fonction du seuil de classification est l'objet de la section suivante.

L'étiquetage des programmes évalué image par image sur la figure 4.10, donne l'impression d'une légère domination de l'ancrage souple sur l'ancrage restrictif, avec un écart d'environ 1% en faveur de l'ancrage souple. Toutefois, les autres schémas montrent que les deux politiques d'ancrage obtiennent des résultats très similaires. Le schéma en haut à gauche de la figure 4.11 montre même que la domination de l'ancrage souple, qui pouvait paraitre constante d'après la figure 4.10 n'est en en fait que ponctuelle, et due à un « accident », lors des journées du 19 et du 20.

En conséquence, les deux politiques d'ancrage semblent se valoir. Les résultats légèrement meilleurs de l'ancrage souple, et sa plus grande simplicité, nous conduisent à la définir comme la politique par défaut pour la suite des travaux.

4.6.4 Résultats en fonction du seuil de classification P/IP

Nous revenons, dans cette section, sur l'influence du seuil de classification P/IP sur les résultats, en ne considérant que l'ancrage souple, qui est désormais la méthode par défaut. Le seuil de classification P/IP permet de faire varier le rappel et la précision de la segmentation de manière importante (voir section 3.4.2.1). Le but de cette section est d'étudier l'impact de la variation de ce seuil, et donc la qualité de la segmentation, sur les résultats de l'étiquetage.

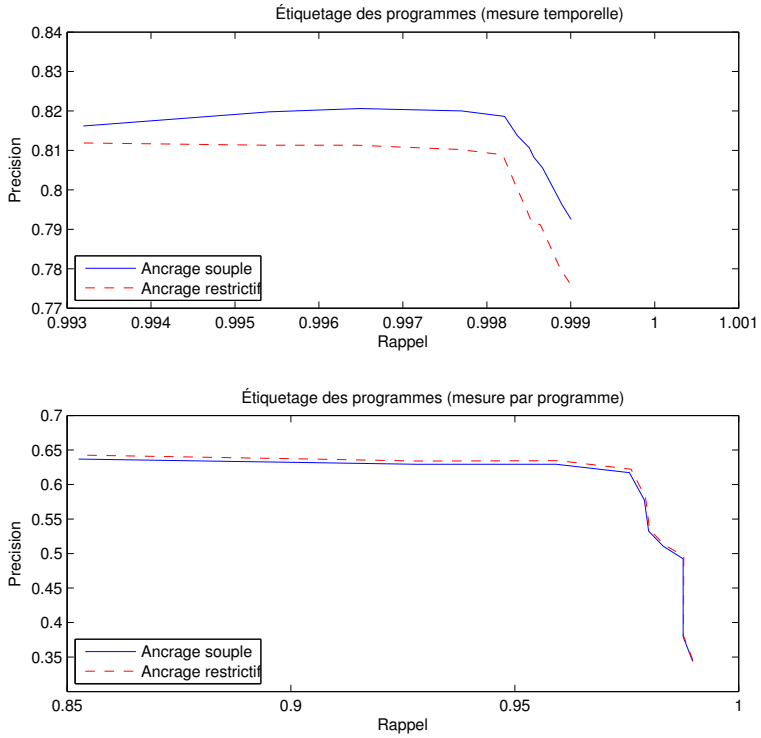

FIG. 4.10 – Courbes précision/rappel de l'ancrage souple et de l'ancrage restrictif, résultats moyennés sur 20 jours.

Les résultats sont visibles sur la figure 4.10, avec la mesure temporelle et la mesure par programme. Ils montrent que la variation du seuil de classification P/IP est aussi cruciale pour la qualité de l'étiquetage. La précision est assez bonne, pour la mesure temporelle, avec environ 82% de bon étiquetage en moyenne sur 20 jours de test. Ce chiffre cache en fait des disparités en fonction des programmes. Les programmes les plus longs de la journée et du début de soirée sont généralement bien étiquetés, car ils sont correctement annoncés dans le guide, et se prêtent bien à un alignement par DTW. Les programmes de la nuit sont beaucoup plus problématiques, souvent à cause d'une mauvaise segmentation, voir la figure 3.15, page 96, à ce sujet. Il existe aussi des difficultés pour les programmes interstitiels, qui sont assez courts, de l'ordre d'1 à 20 minutes, donc difficiles à distinguer des inter-programmes. De plus, ces programmes sont rarement annoncés dans le guide des programmes. La figure 4.12 donne un exemple où cinq programmes consécutifs ne sont pas annoncés par le guide. Dans ce cas, notre seul espoir est la résolution d'ambiguité, à condition que ces programmes possédent un générique, ou un élément répété. Sinon, ces programmes ne peuvent pas être correctement étiquetés, puisque l'étiquette n'est pas connue.

Le chiffre convenable de 82% d'images correctement étiquetées est donc atteint sur-

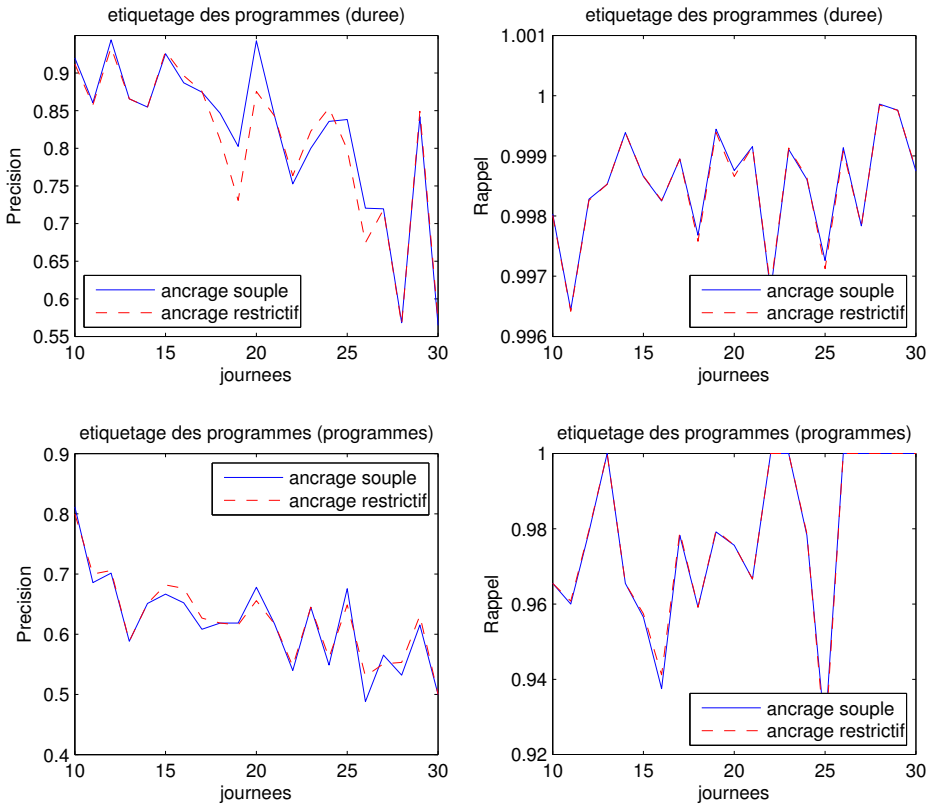

FIG. 4.11 – Comparaison des performances de l'ancrage souple et de l'ancrage restrictif jour par jour.

tout grâce aux programmes longs. La mesure par programmes a, en effet, des résultats beaucoup plus faibles, de l'ordre de 60 à 65% en moyenne, dûs au fait que les programmes courts, qui sont nombreux, sont en général mal étiquetés.

Afin de déterminer la valeur du seuil de classification qui permet le meilleur étiquetage, nous utilisons la même méthode que pour la segmentation, section 3.4.1, c'est à dire que nous cherchons à maximiser la F-mesure. Nous obtenons une valeur très proche de celle obtenue pour la segmentation (1300 images pour la segmentation, 1200 pour l'étiquetage), ce qui est heureux. Ce seuil est fixé en pratique à 1300 images, soit environ 50 secondes.

4.6.5 Apports de la DTW ancrée et de la résolution d'ambiguités

Cette section étudie les apports effectifs de la DTW ancrée et de la méthode de résolution d'ambiguité. La figure 4.13 donne les résultats d'étiquetage en mesure temporelle et en mesure par programme, pour les trois types de méthodes proposées : DTW

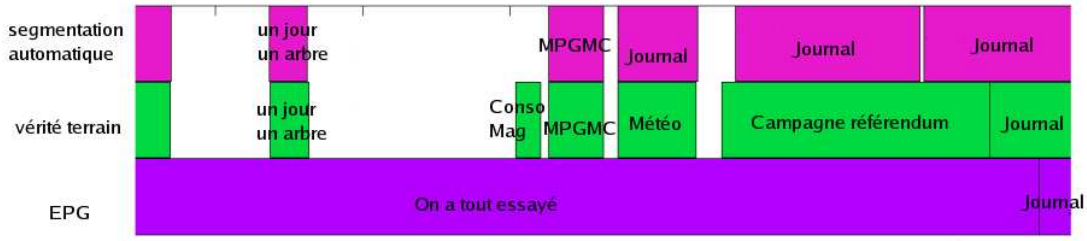

FIG. 4.12 – Exemple de difficultés liées au manque d'information de l'EPG sur un extrait de la journée du 22/05/2005.

classique, DTW ancrée (DTWA sur la figure), et DTWA avec résolution d'ambiguités (DTWA2 sur la figure). Les résultats sont exprimés par la F-mesure, plus compacte que la précision et le rappel, et définie par :

$$F = \frac{2 * précision * rappel}{précision + rappel}$$

Les résultats sont conformes à nos attentes : l'ancrage permet bien une amélioration de l'étiquetage, même si l'amélioration est parfois faible. Il n'existe parfois pas d'ancrage possible, on se ramène donc dans ce cas à une DTW standard.

La résolution d'ambiguité est, quant à elle, assez efficace, ce qui est surtout visible sur le graphe donnant la mesure par programme. La méthode permet, en effet, d'étiqueter un assez grand nombre de programmes interstitiels, ce qui se traduit donc par une forte élévation du score par programme, mais pas forcément du score image par image. La résolution d'ambiguité est en revanche très dépendante du contenu de l'EVR : elle n'a d'intérêt que si les programmes intersticiels sont effectivement reconnus, et donc présents dans l'EVR. Certains de ces programmes ont une durée de vie assez courte, ce qui pose problème, et plaide, encore une fois, pour une mise à jour régulière de l'EVR. On peut remarquer, à ce titre, que l'écart entre DTWA2 et DTW diminue au cours du temps sur la figure 4.13.

4.6.6 Résultats au cours du temps

Nous étudions maintenant les résultats au cours du temps, visibles sur la figure 4.11 en terme de précision/rappel, et sur la figure 4.13 en terme de F-mesure. Un simple coup d'oeil sur les courbes permet de voir que les résultats sont erratiques d'un jour sur l'autre. Cela peut-être étonnant connaissant la grande stabilité de la grille des programmes, qui est en général un parti-pris de la chaîne pour fidéliser le téléspectateur [Dom00, Pol07]. Cette stabilité concerne toutefois essentiellement les programmes majeurs, et ne s'applique visiblement pas aux programmes interstitiels. La grande variation des résultats a principalement deux causes : la première est que la qualité du guide des programmes est elle-même erratique : l'annonce des programmes interstitiels n'est pas stable. La deuxième est que la présence de ces programmes est différente selon la journée considérée, notamment en raison de leur rôle de « variable d'ajustement ».

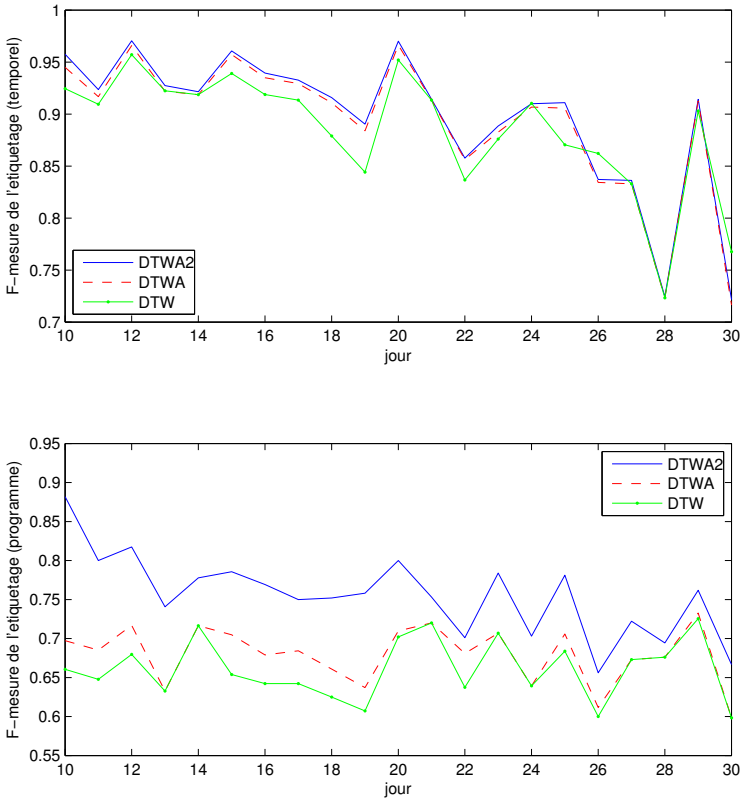

FIG. 4.13 – Comparaison des différentes méthodes proposées pour l'étiquetage sur l'ensemble du corpus 2. DTWA est la DTW ancrée, et DTWA2 est la DTW ancrée avec résolution d'ambiguités.

Le rappel, que ce soit pour la mesure temporelle ou la mesure par programme, est assez peu intéressant, puisqu'il ne mesure que les programmes manqués et non les étiquetages erronés. Il est donc essentiellement lié à la qualité de la segmentation, et à la valeur du seuil de classification P/IP. C'est donc surtout la précision qui caractérise la qualité de l'étiquetage. On s'aperçoit que cette précision est décroissante au fil du temps, de façon non linéaire mais assez rapide puisqu'elle passe de 80% à 50% en seulement 20 jours. Ces résultats ne sont pas étonnant : la section 3.4.2 a montré que la qualité de la segmentation décroissait au cours du temps du fait du non-renouvellement de l'EVR. La segmentation étant moins bonne, l'étiquetage l'est aussi. À segmentation identique, le vieillissement de l'EVR a toutefois assez peu d'impact sur la qualité de l'étiquetage, contrairement à la segmentation. Ceci s'explique par le fait que les programmes se renouvellent beaucoup plus lentement que les inter-programmes, les étiquettes des programmes ont donc une durée de vie plus importante que les inter-programmes. La

décroissance des résultats se retrouve aussi sur la figure 4.13, qui donne les résultats en terme de F-mesure.

La rapide dégradation des résultats, dûe au vieillissement de l'EVR, plaide en faveur d'une méthode de mise à jour de l'EVR. Ceci est l'objet du chapitre suivant.

4.7 Synthèse

Ce chapitre a proposé une méthode d'étiquetage d'un flux de télévision, à partir d'une segmentation en programmes et du guide des programmes. Un alignement global entre la segmentation automatique et le guide des programmes est réalisé en utilisant un algorithme de *Dynamic Time Warping* (DTW).

Nous proposons ensuite une extension de cet algorithme, qui permet de prendre en compte des informations extérieures, apportées par la détection des répétitions du chapitre 2. La prise en compte de cette information de reconnaissance est réalisée grâce à des ancres insérées dans la matrice des coûts de la DTW, de telle sorte que le chemin de la DTW est alors contraint de passer par ces ancres. L'algorithme final est dénommé DTW ancrée.

Nous proposons ensuite une technique de post-traitement, appelée *résolution d'ambiguités*, qui est invoquée lorsqu'il existe une ambiguité sur un segment, c'est à dire qu'il existe dans ce segment une ou des reconnaissances qui ont des étiquettes différentes de celle proposée par la DTW ancrée. Un test d'hypothèse bayésien est utilisé pour déterminer laquelle des deux hypothèses, de la DTW ancrée ou de la reconnaissance, est la bonne.

Nous testons ensuite ces méthodes sur l'ensemble du corpus 2, de 3 semaines de télévision, et nous obtenons des résultats tout à fait statisfaisant. Les problèmes identifiés sont essentiellement liés à la qualité de la segmentation, notamment pour les programmes diffusés la nuit, ainsi qu'aux programmes interstitiels, pour lesquels nous n'avons souvent pas l'étiquette correcte de disponible. Les résultats permettent aussi de remarquer une assez rapide dégradation des résultats au cours du temps, liée au vieillissement de l'EVR. Le chapitre suivant étudie les moyens de mettre à jour l'EVR afin d'obtenir une stabilité de la qualité de l'étiquetage.

Chapitre 5

Structuration dynamique

5.1 Introduction

La majeure critique à l'utilisation d'une méthode de reconnaissance pour détecter les inter-programmes, est la nécessité de maintenir un ensemble de vidéos de référence à jour. Les résultats de la segmentation en programmes, en section 3.4.2.3, page 92, et ceux de l'étiquetage, en section 4.6.6, page 119, ont tout deux montrés la baisse des résultats au cours du temps. Nous appelons **structuration statique** cette méthode de structuration qui utilise un EVR qui ne varie pas au cours du temps. Par opposition, une méthode de structuration qui utilise un EVR variable au cours du temps, est appelée **structuration dynamique**. Nous utiliserons aussi les termes d'EVR statique et d'EVR dynamique, pour désigner un EVR respectivement invariant ou variable au cours du temps.

Le problème de la mise à jour de l'EVR est connu, mais peu de travaux ont cherchés à le résoudre. À notre connaissance, seuls Lienhart *et al.* [LKE97], et Gauch *et al.* [GS06a] se sont penchés sur le problème. Lienhart *et al.* proposent une méthode à base d'heuristiques simples pour inférer de nouveaux segments de publicité. Le principe est qu'un segment inconnu suffisament court, et encadré de deux segments de publicité, est lui aussi de la publicité. Gauch *et al.* ont récemment proposé d'identifier l'apparition de publicités inconnues en détectant les répétitions, avec une méthode de hachage perceptuel, de philosophie identique à celle proposée au chapitre 2. Les segments qui sont adjacents dans la base, et qui se répètent dans le même ordre dans le flux, sont fusionnés pour former une séquence. Cinq descripteurs visuels sont ensuite extraits de ces séquences. À partir de séquences pré-étiquetées en publicité et non-publicité, une classification par la méthode des k-plus proches voisins dans l'espace formé par ces cinq descripteurs est réalisée, permettant alors de classer chaque séquence en tant que publicité ou non-publicité.

Ce chapitre propose une méthode de mise à jour de l'EVR, qui consiste à inférer les inter-programmes inconnus à partir de la segmentation du flux en P/IP. Une méthode naïve est étudiée dans la section 5.3. Nous montrons, dans la section 5.2, que cette approche naïve n'est pas viable, à cause du problème spécifique des bandes annonces,

qui produisent une sur-segmentation. Nous cherchons alors à organiser les éléments inférés, en essayant de les regrouper en *séquences*. C'est l'objet de la section 5.4.1. Nous proposons ensuite, en section 5.4.2, une méthode qui permet de repérer les bandes annonces parmi ces séquences, et de les étiqueter, résolvant ainsi le problème de la sur-segmentation. Différentes méthodes de mise à jour de l'EVR sont proposées, en utilisant tout ou partie des éléments inférés. Les résultats de la structuration en utilisant cet EVR mis à jour sont finalement donnés en section 5.5.

5.2 Principe de mise à jour de l'EVR

La mise à jour de l'EVR consiste à repérer les segments d'inter-programmes inconnus, et à les ajouter à l'EVR. Les segments d'IP inconnus sont détectés à partir de la segmentation du flux, qui permet de les isoler grâce à un encadrement. Le processus global d'étiquetage du flux, incluant la mise à jour, est schématisé sur la figure 5.1.

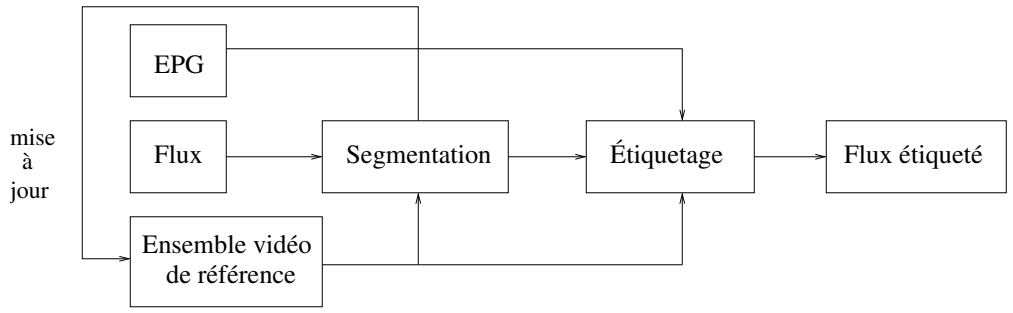

FIG. 5.1 – Schéma du processus global, avec la boucle de mise à jour.

Le principe général est très simple, et semblable à l'approche de Lienhart *et al.* pour les publicités. Un segment dont le type est inconnu a priori est inféré en tant qu'IP si c'est un segment suffisament court, et encadré par deux IP.

Afin de préciser cette notion d'encadrement, nous nous replaçons dans le contexte de la segmentation en programmes et, plus précisément, à l'étape de classification, étape décrite en 3.4.1, page 88. Le premier schéma de la figure 5.2 montre les pré-segments avant l'étape de classification. Nous appelons **segments inférés** les pré-segments qui sont classés en tant qu'IP par le processus de classification. Ces segments inférés sont visibles en vert sur le deuxième schéma de la figure 5.2.

Il est raisonnable de penser que ces segments inférés ne sont pas présents dans l'EVR, puisqu'ils n'ont pas été reconnus. Ces segments sont donc inconnus, et donc pertinents à ajouter à l'EVR afin de le mettre à jour.

5.3 Mise à jour exhaustive

Cette section décrit une méthode de mise à jour dite « exhaustive ». La constatation à l'origine de cette méthode est qu'un segment d'IP inféré par la méthode précédente

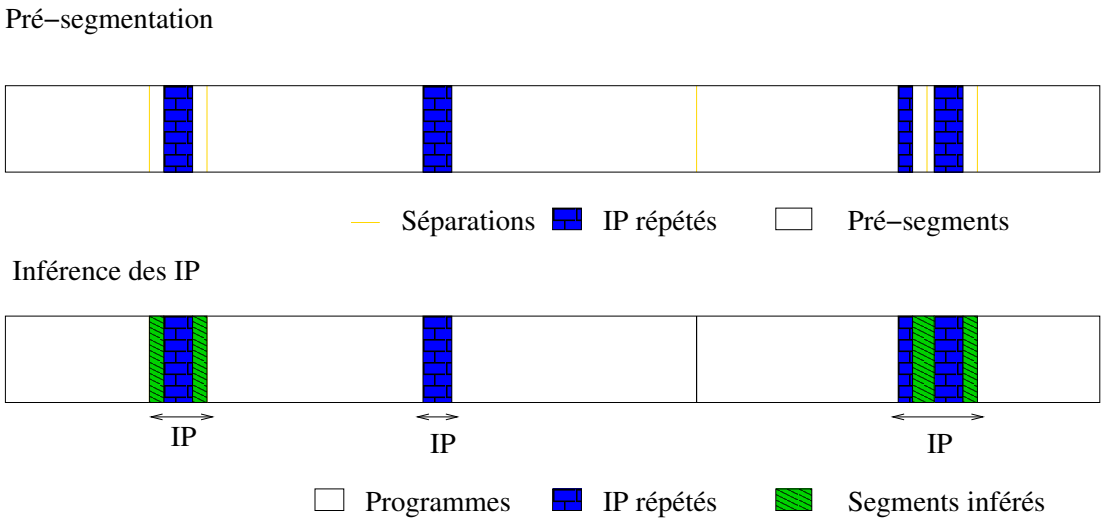

FIG. 5.2 – Inférence de segments d'inter-programme par encadrement.

se répète certainement à d'autres endroits dans le flux. Il est probable que toutes ses occurrences ne soient pas détectées par le principe d'encadrement. Par contre, une fois ajouté à l'EVR, ce segment pourra être détecté en tant que répétition.

Le principe de la méthode consiste à alterner itérativement une phase de détection des répétitions avec une phase de segmentation P/IP. La segmentation permet, en effet, d'inférer de nouveaux segments d'IP, qui, une fois ajoutés à l'EVR, permettent une segmentation plus précise, qui fournit en retour de nouveaux segments d'IP, et ainsi de suite, jusqu'à une convergence du processus. L'algorithme converge lorsque l'on n'infère plus de nouveaux segments, l'EVR est alors stable. L'algorithme 8 résume le processus.

```
EVR, requête : flux vidéo ;

Répéter
    | liste_duplicats = cherche_duplicats(EVR,requête) ;
    | segmentation = segmente_requête(liste_duplicats, requête) ;
    | EVR = mise_a_jour_EVR(EVR, segmentation) ;
jusqu'à ce que (convergence)
```

Algorithme 8: Mise à jour de l'EVR

Cette procédure itérative permet de trouver plus de 98% des IP, (voir tableau 5.1), avec un nombre moyen d'itérations de 5. La figure 5.3 donne une comparaison du nombre de détections de répétitions au cours du temps, avec un EVR statique et un EVR dynamique. Elle montre que le nombre de détections reste à peu près stable au cours du temps avec un EVR dynamique, alors qu'il diminue très rapidement avec un EVR

statique.

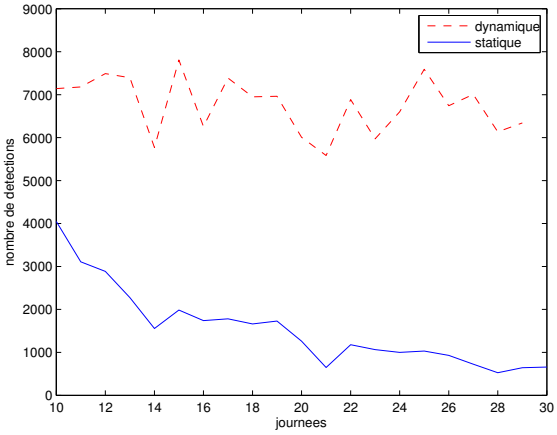

FIG. 5.3 – Comparaison entre le nombre de détections de duplicats obtenu avec un EVR statique et un EVR dynamique

Malheureusement, l'algorithme identifie à tort de nombreux segments de programmes comme étant des IP. Ceci est dû à la présence de bandes annonces, dont les plans sont rediffusés de manière éparse dans le programme qu'elles annoncent. Le problème de la présence des bandes annonces non étiquetées dans l'EVR a déjà été évoqué en 3.4.1, page 90, où nous expliquions que ces bandes annonces sont reconnues comme étant des IP lors de la diffusion du programme qu'elles annoncent, et donc le segmentent. Ceci entraine une sur-segmentation, qui conduit le processus de classification à mal classer de nombreux segments de programmes, qui sont à leur tour intégrés dans l'EVR. Le tableau 5.1 montre la perte en terme de segmentation P/IP, par rapport à la structuration statique. La précision et le rappel de chaque type (P/IP) sont donnés en terme de nombre d'images correctement classifiées (même mesure qu'en 3.4.2).

Méthode	Programme		Inter-programme	
	Précision	Rappel	Précision	Rappel
Méthode statique	98.5	99.9	98.7	85.9
Méthode dynamique	99.8	85.9	45	98.5

TAB. 5.1 – Comparaison entre les résultats de la segmentation P/IP sur 20 jours du corpus 2

L'algorithme 8 n'est donc pas utilisable sous sa forme actuelle. Afin d'utiliser la même méthode qu'avec un étiquetage manuel, il faudrait réussir à détecter, parmi les segments inférés, s'ils contiennent des bandes annonces, et trouver leur titre. C'est ce que nous nous proposons de faire dans les sections suivantes en proposant une méthode dynamique que nous qualifions de **parcimonieuse**, puisque seulement une partie des segments inférés sont effectivement ajoutés à l'EVR.

5.4 Mise à jour parcimonieuse

Nous cherchons dans cette section à limiter les problèmes de sur-segmentation liés aux bandes annonces. Le principe général consiste à trier et organiser les segments inférés, avant de les ajouter à l'EVR.

Afin d'éviter cette sur-segmentation, il est nécessaire de se limiter à une seule itération de l'algorithme 8. Ceci ne génère pas de sur-segmentation, et permet de limiter très fortement le risque que des programmes soient présents dans les segments inférés.

Il s'agit ensuite d'analyser ces segments inférés afin de détecter l'éventuelle présence de bandes annonces. La première étape est d'essayer d'organiser ces segments, en tentant d'identifier les *séquences* qu'ils contiennent. Nous utilisons le terme séquence pour désigner un regroupement de plans sémantiquement homogènes. Une bande annonce, une publicité, un jingle sont des exemple de séquences. Une séquence peut n'être constituée que d'un seul plan (ce qui est fréquemment le cas pour les jingles). Les segments inférés sont généralement constitués de plusieurs séquences.

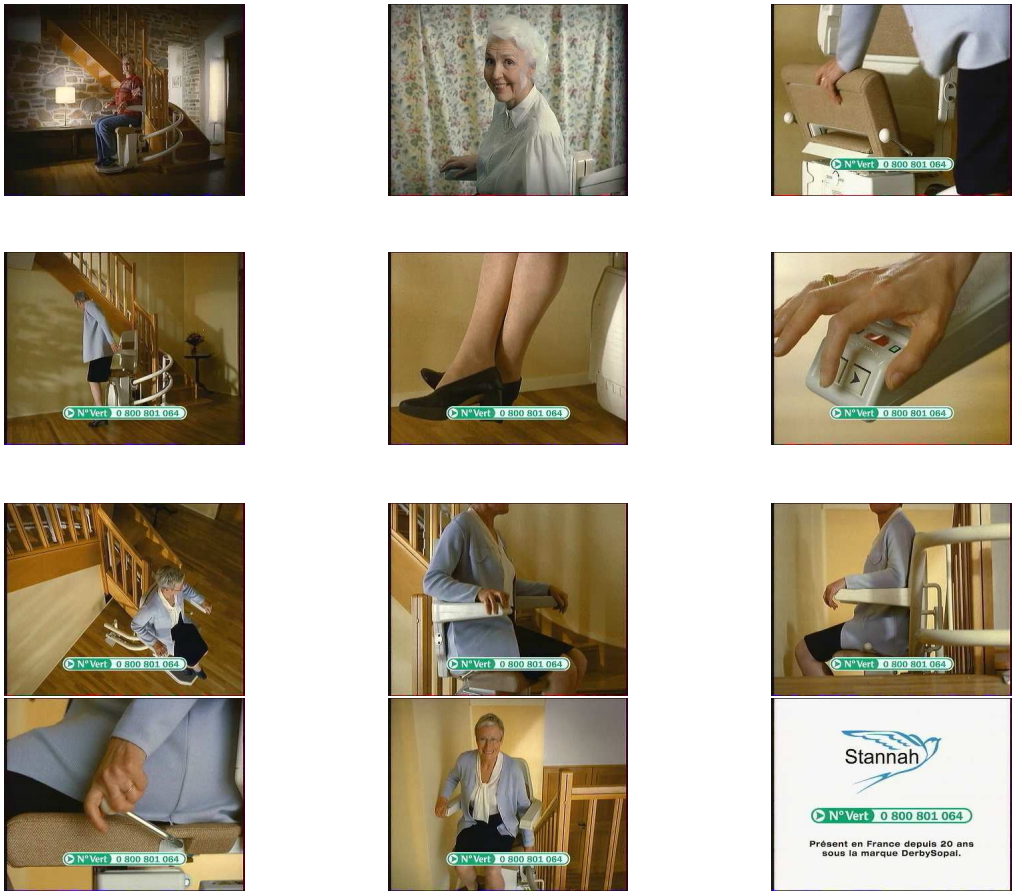

FIG. 5.4 – Exemple d'une séquence composée de 12 plans : une publicité.

La figure 5.4 donne un exemple de séquence de 12 plans, formant une publicité.

On remarque que le deuxième plan est totalement hors contexte (personnage et décor différents), et provient probablement d'une publicité différente. De la même façon, le dernier plan est générique, et est probablement ré-utilisé dans d'autres publicités de la même marque. Ce genre de ré-utilisation peut compliquer la détection des séquences.

L'analyse des segments inférés peut se voir comme une tâche de structuration, avec de la même manière que pour la structuration de flux au niveau du programme, deux grandes étapes : la segmentation, présentée dans la section 5.4.1 et l'étiquetage, en section 5.4.2.

5.4.1 Segmentation en séquences

La segmentation en séquences consiste à segmenter les segments inférés en séquences. Cette tâche peut aussi se voir comme une tâche de regroupement de plans en une séquence.

La segmentation en séquences pourrait se faire à la volée, c'est à dire que pour chaque plan reconnu, le processus teste si le plan précédent a aussi été reconnu, et si les deux plans de l'EVR ont aussi été diffusés consécutivement, auquel cas ils forment alors une séquence. C'est la méthode utilisée par Gauch *et al.*. Cette méthode a l'avantage d'être simple, mais peut, malheureusement, construire des séquences erronées, car elle n'utilise qu'une seule co-occurrence. Afin de limiter les erreurs, nous proposons d'utiliser plusieurs co-occurrences, avant de prendre une décision globale sur les bornes de la séquence. Cette section a pour but d'expliquer comment déterminer ces co-occurrences, en section 5.4.1.1, puis d'expliquer comment déterminer les bornes des séquences, en section 5.4.1.2.

5.4.1.1 Déterminations des voisins

Les plans inférés sont concaténés dans ce que nous appelons l'ESI , l'ensemble des segments inférés. La première étape de la segmentation en séquences, appelée **détermination des voisins**, consiste à parcourir l'historique dans l'ordre temporel, et à déterminer, en utilisant la méthode du chapitre 2, si le plan courant est présent dans l'ESI. Le principe général consiste à repérer les reconnaissances successives dans l'historique, et à construire une liste de voisins, pour chaque plan de l'ESI.

Il existe deux types de voisins : successeurs et prédecesseurs. Définissons la notion de successeur : pour un plan p_i de l'ESI, ayant été reconnu comme répétition d'un plan h_k de l'historique, alors on appelle successeur de p_i le plan p_j de l'ESI qui est reconnu comme répétition de h_{k+1}. Réciproquement, p_j est appelé un prédecesseur de p_i. Un plan de l'ESI peut ne pas avoir de successeur ni de prédecesseur.

L'algorithme précis de la détermination des voisins est donné par l'algorithme 9, et le schéma 5.5 montre de façon plus intuitive la notion de successeur et prédecesseur.

Un problème potentiel est la présence de duplicats dans l'ESI, c'est à dire qu'une même séquence peut avoir été inférée deux fois. Ces duplicats sont nuisibles à la détermination des voisins, car il y a risque de dispersion des voisins entre les duplicats. Le problème est résolu en supprimant les duplicats de l'ESI, avant la détermination des

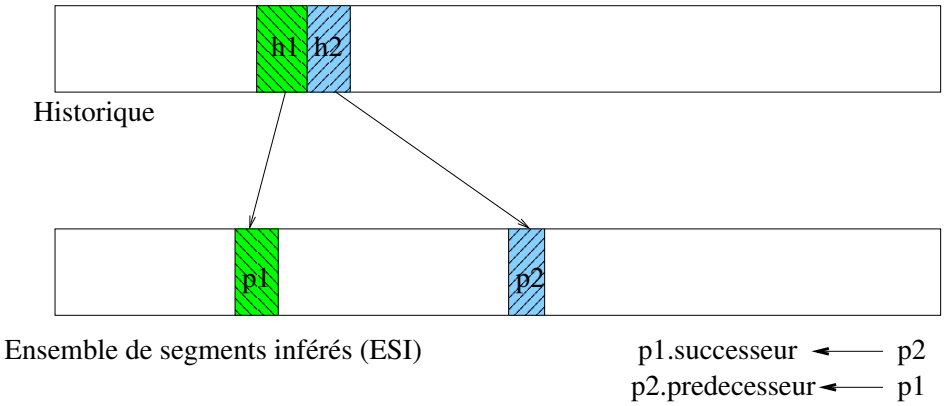

FIG. 5.5 – Illustration de la notion de successeur et de prédecesseur. Ici, p_i est une répétition de h_i, $i = 1, 2$, et p_2 est le successeur de p_1 (resp. p_1 est le prédecesseur de p_2).

voisins, en employant la méthode de détection des répétitions du chapitre 2.

L'historique peut être passé ou futur, il doit par contre bien sûr respecter l'ordre temporel de diffusions des plans. La taille de l'historique, et sa proximité temporelle avec l'ESI (le fait que l'ESI et l'historique ont été diffusés à des dates proches), sont des paramètres important, qui influencent les résultats. Un historique de grande taille et proche temporellement de l'ESI générera, en effet, des statistiques plus pertinentes. En pratique, nous utilisons comme historique l'ensemble du corpus 2, auquel on soustrait la journée qui a servi à construire l'ESI, soit, au final, 20 jours.

En sortie de cette étape de détermination des voisins, un plan p de l'ESI possède désormais une liste de prédecesseurs et de successeurs, ainsi que leurs occurrences. Celles-ci désignent le nombre de fois qu'un plan a été détecté en tant que prédesseur/successeur du plan p.

5.4.1.2 Détermination des bornes des segments

Déterminer les bornes des segments consiste à utiliser les résultats de la détermination des voisins, afin de regrouper les plans qui apparaissent fréquemment ensemble. Cette tache a priori simple s'avère assez délicate, car la détermination des voisins produit des valeurs bruitées. Par exemple, un plan situé au milieu d'une séquence peut ne pas avoir de voisins, parce qu'il n'aura pas été détecté par l'algorithme de détection des répétitions. De la même façon, pour des plans provenant d'une même séquence, les valeurs d'occurrences de leurs voisins peuvent fluctuer, à cause des ratés de la détection des répétitions, ainsi qu'à la présence de duplicats, qui peuvent persister malgré la tentative de nettoyage expliquée en section 5.4.1.1.

Pour résoudre le problème, une analogie avec l'analyse des collocations en traitement automatique des langues est établit. Celle-ci consiste à extraire d'un texte les séquences de mots (ou plus généralement de symboles), qui apparaissent fréquemment ensemble. On définit pour cela des mesures d'association, qui évaluent de manière statistique si

```
Historique = {h_1, ..., h_N1} : Liste de plans ;
ESI : Liste de plans ;
p, p_precedent : plan ;

Pour i de 1 à N_1 faire
    p = detection_répétitions(h_i,ESI) ;
    Si (p ≠ ∅) Alors
        h_i.reco = vrai ;
        Si (h_{i-1}.reco) Alors
            [mise à jour]
            p_precedent.successeurs ← p ;
            p.predeceseurs ← p_precedent;
        Fin Si
        p_precedent = p ;
    Sinon
        h_i.reco = faux ;
    Fin Si
Fin Pour
```

Algorithme 9: Détermination des voisins.

les co-occurrences sont significatives. Une mesure souvent employée est l'information mutuelle [CH89], qui pour deux mots x et y est définie par

$$I(x,y) = \log_2 \frac{p(x,y)}{p(x)p(y)}$$

Cette notion de collocation se transpose aisément à notre problème, où les symboles sont des plans. Si le principe est le même, l'estimation des co-occurrences est cependant différente : l'alphabet, c'est à dire l'ensemble des symboles possibles, est infini et variable au cours du temps. Cette caractéristique impose une estimation en ligne des statistiques de co-occurences, contrairement, par exemple, au cas où les symboles sont des lettres ou des mots, où l'on peut envisager une estimation qui soit valable pour un ensemble de documents. D'un autre côté, le fait que l'alphabet soit infini facilite l'analyse des collocations, car la probabilité *a priori* que deux symboles apparaissent côte à côte est faible. Les statistiques de co-occurrence sont donc plus significatives.

Le problème d'unités atomiques très fréquentes (prépositions...) qui faussent les estimations n'existe pas non plus. Par contre la plupart des modèles statistiques proposés en traitement automatique des langues (TAL) sont généralement restreints à l'estimation de bigrammes, ce qui est problématique dans notre cas car les séquences que nous souhaitons identifier peuvent atteindre des tailles importantes (une bande annonce peut contenir de l'ordre de 30 à 40 plans). Des travaux cherchant à extraire des unités lexicales complexes (au delà du bigramme) existent [GDPL00] mais se concentrent sur les

difficultés qu'engendrent la présence d'unité atomiques fréquentes, ce qui n'a pas de sens dans notre contexte.

Pour intégrer cette notion de séquence à la méthode, nous modélisons l'ESI comme une concaténation de séquences générées par une chaine de Markov d'ordre 1. Un état x_t est un plan de l'ESI, caractérisé par un ensemble de successeurs $S(x_t) = \{(x_{k_i}, \beta_t^{k_i})\}_{i=1...n_\beta^t}$ et de prédecesseurs $P(x_t) = \{(x_{p_i}, \alpha_t^{p_i})\}_{i=1...n_\alpha^t}$. Le scalaire $\alpha_t^{p_i}$ représente le nombre d'occurrences de l'état x_{p_i} en tant que prédecesseur de l'état courant x_t. De la même manière, $\beta_t^{k_i}$ représente le nombre d'occurrences de l'état x_{k_i} en tant que successeur de l'état x_t. L'espace d'état est l'ensemble des plans de l'ESI.

Suivant ces définitions, le nombre de prédecesseurs différents pour l'état x_t est n_α^t et le nombre total de prédecesseurs est

$$N_\alpha^t = \sum_{i=1}^{n_\alpha^t} \alpha_t^{p_i}$$

On définit de la même manière le nombre total de successeurs par

$$N_\beta^t = \sum_{i=1}^{n_\beta^t} \beta_t^{k_i}$$

Les probabilités de transitions entre états sont définies de la manière suivante :

$$p(x_t|x_s) = \begin{cases} \frac{\alpha_t^s + \beta_s^t}{N_\alpha^t + N_\beta^s} & \text{si } s = t-1 \\ 0 & \text{sinon} \end{cases}$$

Cette définition des probabilités de transition implique que les états sont nécessairement dans l'ordre temporel. La chaine de Markov résultante est donc une chaine gauche-droite, sans bouclages, comme le montre la figure 5.6.

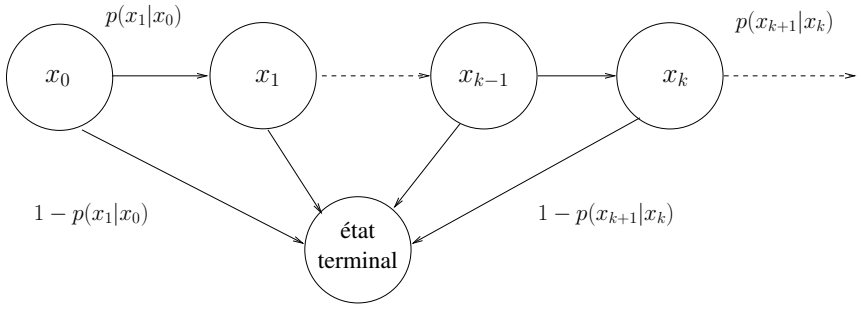

FIG. 5.6 – Chaine de Markov gauche-droite pour la segmentation en séquences

Chaque état peut alors passer dans l'état suivant ou alors passer dans l'état terminal avec une probabilité $1 - p(x_t|x_{t-1})$, ce qui termine alors la séquence. Cette propriété de cohérence temporelle est imposée par le problème fixé qui est non pas le problème traditionnel d'identifier le modèle qui a généré une séquence, mais d'identifier des séquences

à l'intérieur de l'espace des états, sachant que cet espace forme une suite temporelle. En reformulant, il s'agit de segmenter l'espace d'états en une suite de séquences ayant été générées par un processus Markovien.

Pour cela, on parcours l'espace des états dans l'ordre temporel, et on initialise un début de séquence en seuillant la probabilité de transition $p(x_t|x_{t-1}) > \gamma$. Cette étape est connue en TAL sous le nom d'amorçage. Une fois le processus amorcé, une première solution serait de décider que le vecteur $(x_0, \ldots, x_k)$ forme une séquence si $p(x_0 \ldots x_n) > \gamma^{n+1}$. Ce qui est facilement calculable puisque d'après la propriété Markovienne on a :

$$p(x_0 \ldots x_n) = p(x_0) \prod_{k=1}^{n} p(x_k|x_{k-1})$$

en passant au logarithme pour faciliter le calcul, la décision deviendrait :

$$\log p(x_0) + \sum_{k}^{n} \log p(x_k|x_{k-1}) > (n+1) \log \gamma$$

Cette décision est bien adaptée pour décider si le vecteur $(x_0, \ldots, x_k)$ a bien été produit par le processus Markovien correspondant. En revanche, pour notre problème de segmentation, cette décision n'est pas assez précise car elle considère l'ensemble de la séquence et n'est pas capable de prédire l'instant exact de rupture. On la remplace par une décision à chaque instant t au lieu d'une décision globale, en seuillant simplement $p(x_t|x_{t-1}) > \gamma$, ce qui revient à appliquer la condition d'amorçage à chaque instant, et d'associer les plans 2 à 2. Une règle a priori sur le nombre d'hypothèses peut être éventuellement ajoutée afin d'augmenter le rappel, un nombre d'hypothèses élevé étant signe d'une rupture. La décision finale deviendrait alors :

$$(p(x_t|x_{t-1}) > \gamma) \ \textbf{et} \ \left(n_\alpha^t + n_\beta^{t-1} < \delta\right)$$

En pratique, les seuils sont fixés à $\gamma = 0.1$ et $\delta = 6$.

On peut remarquer que la méthode se rapproche finalement assez de la méthode « classique » employée en TAL, qui revient à un seuillage de la mesure d'association pour chaque couple. A des fins de comparaisons, nous appliquons aussi la méthode classique de Church et Hanks [CH89], légèrement ré-adaptée à notre situation. Avec N_T le nombre total de plans dans l'historique utilisé pour l'estimation[1], on a

$$p(x) = \frac{N_\alpha^x + N_\beta^x}{2N_T} \text{ et } p(y) = \frac{N_\alpha^y + N_\beta^y}{2N_T}$$

l'information mutuelle devient :

$$I(x,y) = \log_2 \frac{2N_T(\alpha_x^y + \beta_x^y)}{(N_\alpha^x + N_\beta^y)^2}$$

La méthode de formation d'une séquence est alors d'agglutiner les plans tant que la mesure d'association ne descend pas en dessous d'un certain seuil.

[1] On se restreint donc ici à un alphabet de taille finie, donné par l'historique.

5.4.1.3 Résultats

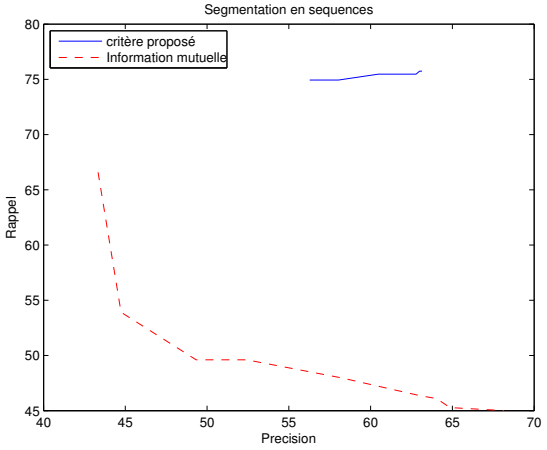

FIG. 5.7 – Résultats de structuration en séquence

L'évaluation est faite sur la journée du 09/05/2005 du corpus 2, sur laquelle on construit un ESI. Une vérité terrain est construite sur cet ESI, qui consiste à donner un nom à chaque plan. Les plans consécutifs de même nom forment alors une séquence. La structuration en séquences est évaluée de la même manière qu'une segmentation en plans. On évalue en terme de rappel et de précision les instants de ruptures. La figure 5.7 présente les résultats, où nous donnons aussi, à titre de comparaison, une segmentation par le critère de l'information mutuelle.

Les problèmes proviennent de plusieurs sources. Nous avons déjà évoqué les faiblesses de la méthode de détection des répétitions, ainsi que la présence de duplicats. Un autre problème est le parrainage, qui est accolé au programme qu'il annonce, et n'est donc pas détectable en tant que séquence propre par la méthode proposée. Un autre cas concerne les bandes annonces à « usage unique », qui sont des pré-annonces de programmes, diffusées une seule fois, juste avant celui-ci. Il n'est pas possible de regrouper ces plans en une séquence, car la bande annonce ne se répète pas.

La courbe montre une nette domination du critère proposé par rapport à l'information mutuelle, et montre aussi le peu d'influence de la valeur du seuil γ (le seul à varier sur la courbe), puisque les résultats sont très stables. Les seuils γ et δ sont choisis de façon à favoriser le rappel sur la précision, et donc de favoriser une sur-segmentation des séquences, car celle-ci a peu d'impact sur la structuration finale. Une sous-segmentation est, au contraire, très peu souhaitable car elle lie entre eux des segments sans rapport, et peut s'avérer catastrophique, notamment dans la méthode d'étiquetage développée dans le paragraphe suivant.

5.4.2 Étiquetage de séquences inconnues

5.4.2.1 Problématique et algorithme général

Cette section a pour but d'essayer d'étiqueter les séquences obtenues par la méthode de la section précédente, à partir des informations fournies par le guide des programmes. L'étiquetage résultant fournit une information extrêmement utile au processus de structuration, semblable à l'information qu'apporte une segmentation et un étiquetage manuel.

Nous repartons donc des séquences inférées. Un premier traitement très simple est d'utiliser la méthode de Lienhart [LKE97] qui consiste à inférer qu'une séquence est une publicité si elle est encadrée par deux publicités, et si elle est suffisament courte (de l'ordre d'une dizaine de secondes). Seul le genre est inféré, pas le titre, mais cela est équivalent à l'étiquetage manuel employé (pas de différenciation des publicités par leur nom).

```
EVR, historique, requête : flux vidéo ;
liste_sequences : liste[sequence] ;
s : sequence ;
p, copie_de_p : plan ;

structuration = structuration_statique(EVR,requête) ;
Pour chaque s dans liste_sequences faire
    Pour chaque p dans s faire
        copie_de_p = detection_répétitions(p,historique) ;
        Si (copie_de_p ≠ ∅) Alors
            Si ((copie_de_p ∩ structuration) est de type programme) Alors
                Si (Not est_une_sequence(copie_de_p,s)) Alors
                    [ici se trouve le programme annoncé par la séquence s]
                    [s prend donc l'étiquette du programme trouvé]
                    s.etiquette = (copie_de_p ∩ structuration).etiquette ;
                    break
                Fin Si
            Fin Si
        Fin Si
    Fin Pour
Fin Pour
```

Algorithme 10: Étiquetage de séquences inconnues

Un deuxième traitement se concentre sur les bandes annonces. L'idée générale est de détecter la localisation, dans le flux, du programme annoncé par la bande annonce. Cette détection du programme annoncé est possible, car une bande annonce contient généralement des plans du programme à venir, à de légères transformations près. Ces

transformations comprennent des insertions de cadres, de texte, de logos et parfois un redimensionnement de l'image originale. Le descripteur utilisé ici n'est pas toujours capable d'absorber ces transformations. Pour plus de robustesse, une méthode inspirée des travaux de Joly [Jol05] serait peut être plus appropriée. L'algorithme 10 détaille le processus général utilisé. La méthode se décompose en deux parties : un pré-étiquetage, et la recherche du programme annoncé.

Le pré-étiquetage consiste à étiqueter un historique, en réalisant une structuration statique de cet historique. Cela nécessite un EVR statique, qui sert à « amorcer » le processus de mise à jour. Notons que l'historique doit contenir le futur, puisqu'il doit contenir le programme annoncé par la bande annonce. La méthode a donc besoin d'un différé assez important pour fonctionner, de l'ordre de quelques jours, voir plus d'une semaine pour certains programmes.

La recherche du programme annoncé consiste à parcourir l'historique, en détectant les répétitions des plans de la séquence. Le point crucial est de décider, lorsqu'un plan de la séquence est reconnu, si ce plan est isolé, ou si ce plan apparait dans le contexte de la séquence auquel il appartient. Si ce plan est isolé, alors c'est qu'il apparait dans le contexte du programme et non dans celui d'une bande annonce. C'est ce que fait la fonction *est_une_sequence* détaillée dans la section suivante.

5.4.2.2 Détections de plans isolés

Nous cherchons à déterminer si le plan reconnu dans l'historique fait partie d'une séquence, ou bien est une détection isolée.

D'après l'algorithme 10, nous sommes en possession d'un plan p qui a été reconnu comme étant une répétition d'un des plans, appellons-le s_k, de la séquence requête S. L'algorithme retenu consiste simplement à aligner S avec une portion de l'historique de même taille, l'alignement étant fourni par les positions de p avec s_k. Cet alignement est illustré par la figure 5.8. Remarquons que cette méthode d'alignement est exactement la même que pour la distance ancrée, la différence est que ce sont des plans et non des signatures que nous alignons ici.

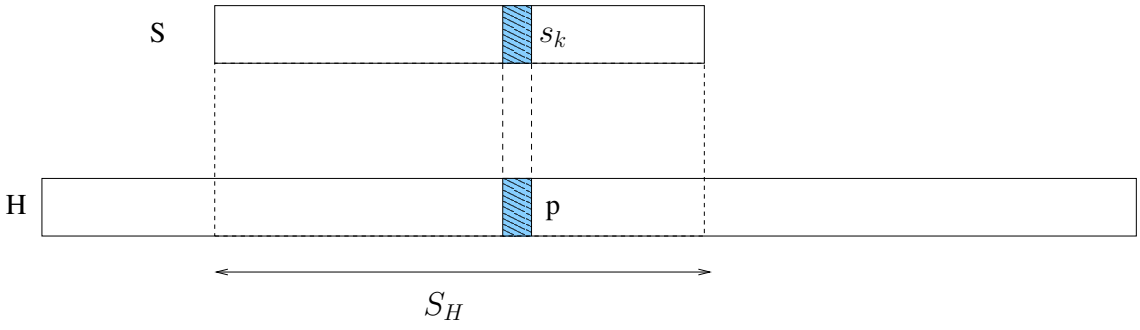

FIG. 5.8 – Illustration de l'alignement d'une séquence S sur l'historique H, grâce à la détection du plan s_k comme étant une répétition du plan p.

Une fois l'alignement effectué, il s'agit ensuite de définir une distance entre les sé-

quences. Deux types de méthodes peuvent être envisagés. Le premier type est de concaténer l'ensemble des descripteurs de la séquence requête en une suite temporelle de descripteurs et de chercher cette suite dans la portion considérée de flux. La distance entre les suites de descripteurs peut être une simple distance de Hamming moyenne ou une distance d'édition. Les deux séquences sont déclarées identiques si la distance est en dessous d'un certain seuil.

Le deuxième type de méthode envisagé est de conserver le découpage en plan et de calculer une distance plan par plan. Nous retrouvons dans ce cas la problématique de détection de répétition de plans, présentée en section 2.4.2, où l'étude faite a montré qu'une distance de hamming correctement alignée grâce à une position exacte d'un descripteur (une ancre) est la meilleure solution. Il suffit donc de tester l'égalité de chacun des plans des deux séquences, S et S_H. La sortie de cet algorithme est le pourcentage de plans identiques entre S et S_H. Une décision est prise à ce niveau du procesus, en décidant que si la séquence S partage moins de 80% de plans en commun avec la séquence S_H, alors S_H n'est pas une répétition de la séquence S. Ce seuil de 80% a été fixé empiriquement, et n'est pas critique. Ce seuil doit être supérieur au pourcentage de plan diffusés consécutivement à la fois dans la bande annonce et dans le programme. Les bandes annonces sont, en général, montées spécifiquement pour donner un tythme, et ne sont pas juste un extrait du programme. Nous n'avons pas d'exemple, dans notre corpus, de bande annonce sans découpage. Le seuil doit, par contre, être inférieur à un pourcentage maximal, afin de prendre en compte d'éventuels ratés du processus de détection de répétition. En résumé, une valeur de seuil comprise entre 60 et 90% peut être choisie, sans grand changement dans les résultats.

5.4.2.3 Décision

Nous détaillons ici le principe de décision concernant l'étiquetage de la séquence. L'algorithme 10 est en fait simplifié dans sa prise de décision puisqu'il se contente d'étiqueter la séquence dès qu'une hypothèse valide est trouvée. Cette solution n'est pas utilisable en pratique car des faux positifs peuvent apparaitre dans la détection des plans isolés, mais surtout l'étiquette inférée (qui vient de la structuration statique) peut être erronée, la structuration statique n'étant pas parfaite. Pour limiter les erreurs, l'algorithme parcourt l'ensemble de l'historique afin de collecter un ensemble d'hypothèses, et leur nombre de votes. Un vote est obtenu pour chaque instance de plan reconnu comme étant isolé (c'est à dire n'appartient pas à la séquence d'origine). Pour un ensemble d'hypothèses et leurs votes respectifs (h_i, v_i) la décision est prise d'étiqueter la séquence avec l'hypothèse h_k , $k = \arg\max_i v_i$ si :

$$\frac{v_k}{\sum_i v_i} > \alpha \text{ et } v_k \geq 2$$

Le seuil α est en pratique fixé à $\frac{1}{2}$. La contrainte $v_k \geq 2$ n'est pas nécessaire mais permet de filtrer un certain nombre de faux positifs ponctuels, puisqu'ils n'ont qu'un seul vote. En revanche, cette condition supprime aussi un certain nombre de bandes annonces correctement identifiées mais qui n'apparaissent qu'une seule fois et dont un seul plan

est reconnu. Cette condition peut donc éventuellement être supprimée si l'on souhaite un taux de rappel plus élevé, et si la suite du processus est résistante aux étiquetages erronés.

5.4.2.4 Résultats partiels

Cette section teste la validité de la méthode d'étiquetage des bandes annonces et essaye d'évaluer les taux de bon étiquetage. Pour cela, l'algorithme d'étiquetage n'est pas appliqué sur des séquences inférées, car il existe alors des problèmes de segmentation, dus à une mauvaise segmentation en séquences. De plus l'ensemble des bandes annonces ne sont pas inférées par la méthode précédente, et les résultats peuvent donc être biaisés. Afin de ne tester que la validité de l'algorithme d'étiquetage de bandes annonces, le protocole expérimental proposé est tout simplement de prendre les bandes annonces d'une journée du corpus 2, dont les positions sont connues grâce à la vérité terrain, et d'essayer de les étiqueter.

TAB. 5.2 – Résultats de l'étiquetage de séquence

Méthode	Précision	Rappel	F-mesure
Distance d'édition signature	99.7	63.7	77.7
Hamming signature	100	58.3	73.7
Hamming plan	93.4	66.1	77.4

Plus précisément, pour une journée k du corpus 2, toutes les bandes annonces sont extraites à partir de la vérité terrain. Ces bandes annonces forment des séquences, c'est à dire une suite de plans, que nous essayons d'étiqueter avec l'algorithme d'étiquetage. L'historique est choisi comme étant le regroupement des journées de $k+1$ à $k+6$, c'est à dire 5 jours de flux.

L'étiquetage est évalué par plan, c'est à dire que pour chaque plan des bandes annonces présentées en entrée de l'algorithme d'étiquetage, on évalue en sortie si le plan est correctement étiqueté ou non, ou alors n'a pas été étiqueté. Nous pouvons alors exprimer les résultats en terme de précision et rappel. La table 5.2 présente les résultats de cet étiquetage, moyennés sur 4 jours, du 9/05/2005 au 12/05/2005. Les trois méthodes présentées en section 5.4.2.2 sont données dans cette table : deux méthodes par concaténation, une avec une distance de Hamming moyenne entre les signatures, et l'autre par distance d'édition. La troisième est la méthode basée plan. D'après les chiffres, c'est la méthode par concaténation avec distance d'édition qui semble la meilleure. En fait, une analayse plus détaillée des résultats montre que c'est la distance de Hamming par plan qui est la plus efficace pour repérer les bandes annonces dispersées. Ses résultats sont inférieurs à cause d'un mauvais étiquetage de l'historique. Nous rappelons ici que l'historique est étiqueté par une structuration statique, qui est donc sujette à erreurs. Le processus peut donc trouver le programme effectivement annoncé par la bande annonce, mais qui est mal étiqueté par la structuration statique. Dans nos expérimentations, ce cas de figure n'est apparu que pour des programmes diffusés la nuit, pour lesquels la structuration est, effectivement, souvent erronée.

Un problème relativement important se produit lorsque deux bandes annonces différentes comportent un ou plusieurs plan(s) commun(s). Ce cas se produit plusieurs fois dans notre corpus, par exemple, dans le cas d'un film annoncé par plus de 30 bandes annonces différentes, qui possédent toutes quelques plans communs. Ce cas est très défavorable, car l'occurrence dans l'historique d'une bande annonce « cousine », sera interprétée par notre algorithme comme étant le film lui-même. Le critère de décision permet, en général, de filtrer ces erreurs, et donc de ne pas générer un étiquetage erroné, mais ce phénomène peut faire diminuer le rappel, car il produit un grand nombre d'hypothèses. Les autres cas d'erreurs sont dus à des bandes annonces qui ne respectent pas les hypothèses faites, par exemple le cas des multi annonces, plusieurs programmes différentes annoncés en même temps, ou le cas de bandes annonces dont une grande partie des plans appartiennent à un programme déjà diffusé (cas d'un fim en deux parties, mais à bandes annonces fixes).

5.4.3 Méthodes de mise à jour

À la suite de ces traitements, les segments inférés sont alors divisés en 3 types :

1. les segments regroupés en séquences et étiquetés (type 1)
2. les segments regroupés en séquences mais non étiquetés (type 2)
3. les segments isolés inconnus (type 3)

Ces segments sont utilisés pour définir différentes stratégies de mise à jour. La première stratégie est de n'ajouter à l'EVR que les segments de type 1. C'est la solution la plus conservative, qui permet de garder un EVR de qualité, c'est à dire entièrement étiqueté, au détriment du nombre de segments inférés. Les segments de parrainage et de publicités, qui ne peuvent pas être étiquetés par la méthode proposée en 5.4.2 ne sont en effet jamais pris en compte, et une dégradation inévitable des résultats apparaitra donc au fil du temps. La deuxième stratégie est d'ajouter à l'EVR les segments de type 1 et 2, ce qui permet d'augmenter considérablement le nombre de segments ajoutés à l'EVR. Ceci se fait parfois au prix d'une sur-segmentation, lorsqu'il n'a pas été détecté qu'une séquence était en fait une bande annonce et n'a donc pas été étiquetée. Elle génère donc l'effet de sur-segmentation décrit précédemment (section 5.2). La troisième stratégie consiste à incorporer l'ensemble des segments dans l'EVR, autrement dit, les types 1, 2 et 3, ce qui permet d'inclure l'ensemble des segments inférés dans l'EVR, mais au prix d'une certaine perte de qualité dans l'étiquetage final, les segments isolés non étiquetés pouvant introduire des perturbations fortes sur le processus global d'alignement et d'étiquetage.

5.5 Résultats

La méthode de structuration dynamique parcimonieuse nécessite un différé, qui correspond idéalement au temps entre lequel la première bande annonce est diffusée, et le moment où l'émission annoncée est effectivement diffusée. Plus ce différé est important, et plus le processus a de chances de trouver l'ensemble des émissions annoncées, et donc

d'améliorer les résultats. Cependant, un long différé rend le procesus plus complexe et peu pratique. Un différé de 5 jours est utilisé pour l'ensemble des résultats présentés ici, ce que l'on considère comme un bon compromis entre le rappel et la complexité.

Par défaut, la méthode de mise à jour utilisée dans la méthode dynamique parcimonieuse est de type 2 : seuls les nouveaux segments étiquetés, ou sous forme de séquence, sont ajoutés à l'EVR. Les performances des différents types de mise à jour sont analysées dans la section 5.5.4.

5.5.1 Comparaison des méthodes statique et dynamique parcimonieuse

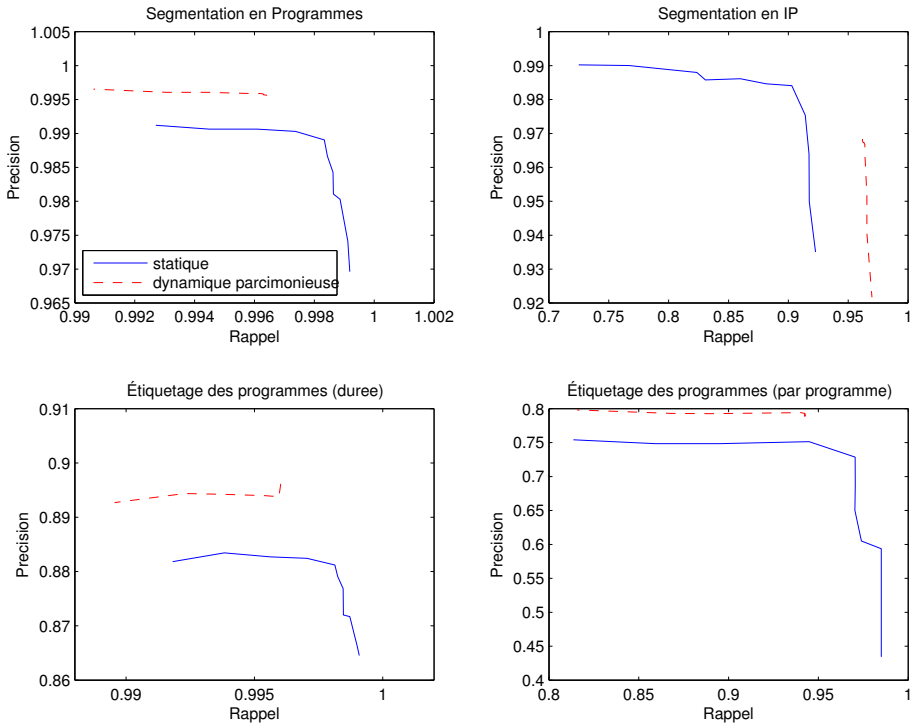

FIG. 5.9 – Comparaison des méthodes statique et dynamique parcimonieuse pour la segmentation P/IP et l'étiquetage sur un corpus de 120 heures

La figure 5.9 donne les résultats en terme de rappel et précision, respectivement, pour la segmentation en P/IP, et pour l'étiquetage, et ceci pour les méthodes statique et dynamique parcimonieuse. Ces résultats sont obtenus sur une durée de corpus de 5 jours, soit 120 heures. Le paramètre crucial qui est utilisé pour faire varier les résultats en précision/rappel sur les figures, est le seuil de classification P/IP. Cette figure montre que, la méthode de structuration dynamique parcimonieuse possèdant un EVR plus complet, le rappel de la segmentation en IP est naturellement meilleur qu'en statique,

mais la précision est aussi moins bonne (et réciproquement pour la segmentation en programmes). Le même phénomène affecte les résultats de l'étiquetage.

La bonne tenue des résultats en statique s'explique par le fait que l'EVR est proche temporellement des journées de tests, ce qui génère un grand nombre de reconnaissances et donc une bonne segmentation.

5.5.2 Étude de l'influence de la détection des séparations

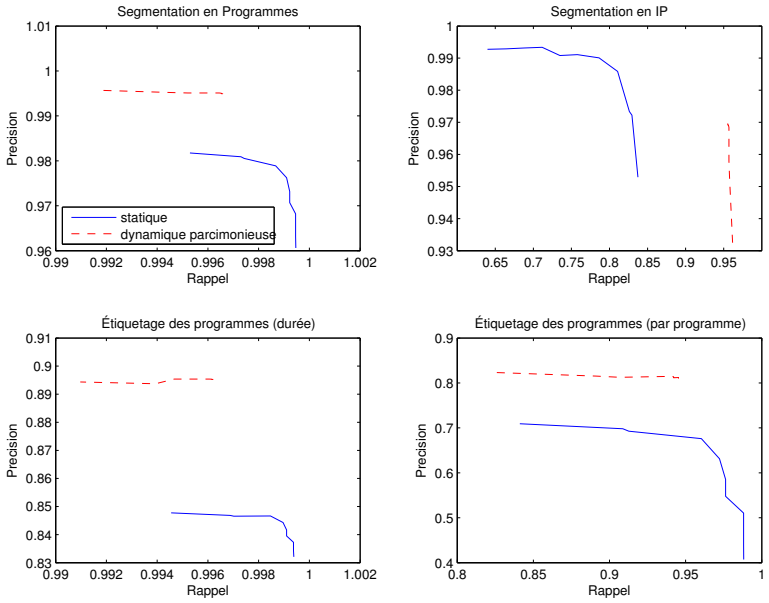

FIG. 5.10 – Résultats de l'étiquetage des programmes sans détection des séparations

Une des affirmations du chapitre 3 était que la détection des **séparations** entre publicités par le biais des images monochromes et du silence n'était pas indispensable. La figure 5.10 montre les résultats de segmentation P/IP et d'étiquetage pour les méthodes statiques et dynamique parcimonieuse **sans** cette détection des séparations. Ces courbes, comparées à celles de la figure 5.9, confirment l'intérêt des séparations, lorsque la méthode statique est utilisée, car en leur absence on observe une forte chute des résultats (par exemple le rappel sur la segmentation en IP). L'intérêt majeur de ces courbes est, toutefois, la performance de la méthode dynamique parcimonieuse, qui montre ici une nette supériorité sur la méthode statique, et qui obtient des performances à peu près équivalentes, que l'on utilise les séparations ou pas. Ces résultats confirment donc la possibilité de n'utiliser que les reconnaissances pour la structuration, à condition d'une mise à jour régulière et de qualité de l'EVR.

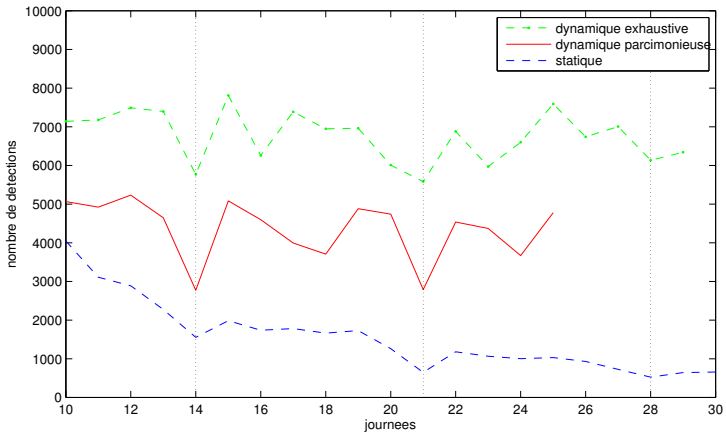

FIG. 5.11 – Comparaison du nombre de détections de répétitions pour les 3 types de méthodes de structuration : statique, dynamique et dynamique parcimonieuse. Les lignes verticales en pointillées indiquent les dimanches.

5.5.3 Résultats au cours du temps

La figure 5.11 montre le nombre de détections entre la journée requête et l'EVR au cours du temps, avec les différentes méthodes de mise à jour de l'EVR : statique (pas de mise à jour), dynamique (procédure expliquée dans la section 5.2), et dynamique parcimonieuse, avec une méthode de mise à jour de type 2. Les méthodes dynamiques montrent une relative stabilité du nombre de détections au cours du temps, alors que le nombre de détections baisse très rapidement avec un EVR statique. Comme attendu, le nombre de détections avec un EVR dynamique parcimonieux est légèrement inférieur à celui obtenu par la méthode dynamique exhaustive. À titre de curiosité, la figure marque par des lignes pointillées les journées du 14, 21, et 28, qui accusent une forte chute du nombre de détections, quelque soit la méthode. Ces journées sont, en fait, des samedis/dimanches[2], qui possèdent une programmation très différente des journées de la semaine, ce qui se traduit par une baisse du nombre de répétitions. À noter aussi que la courbe de la méthode dynamique parcimonieuse s'arrête à la journée du 25, puisqu'elle nécessite un différé, égal ici à 5 jours.

Le nombre important de détections des méthodes dynamiques montre seulement que de nombreux plans sont inférés, mais ne donne aucune garantie sur la qualité de cette inférence. La figure 5.12 donne les résultats en terme de segmentation pour les différentes méthodes, et montre les mauvais résultats obtenus par la méthode dynamique exhaustive. Le rappel de la segmentation en IP (réciproquement la précision de la segmentation en programmes) est excellent, puisque la méthode exhaustive infère la presque totalité des nouveaux IP. En revanche, la méthode infère aussi des segments qui

[2]Nous rappelons que nos journées de test sont à cheval entre deux journées, puisqu'elles commencent à 15h28.

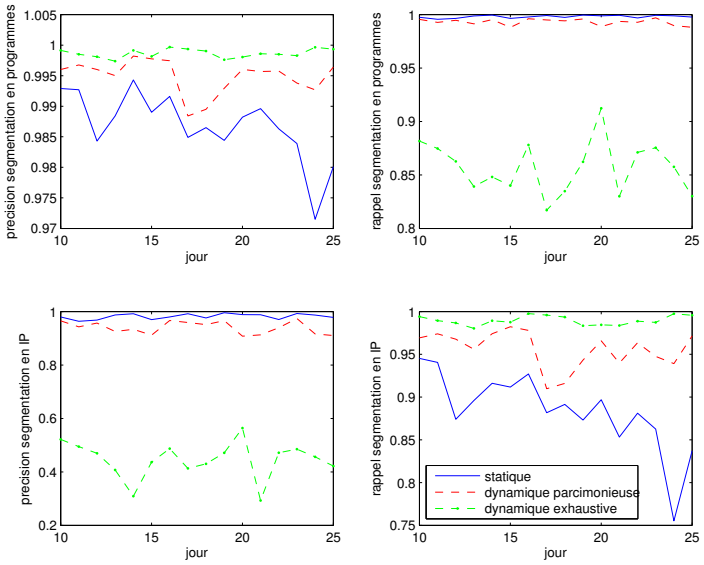

FIG. 5.12 – Comparaison des résultats de segmentation P/IP obtenus par les méthodes statique, dynamique exhaustive, et dynamique parcimonieuse.

ne sont pas des IP, ce qui génère le phénomène de sur-segmentation. Avec une précision moyenne de la segmentation en IP de l'ordre de 0.5, la méthode dynamique génère une segmentation qui n'est pas exploitable.

La figure 5.12 montre, en revanche, le réel apport de la méthode dynamique parcimonieuse. Au prix d'une très légére perte en précision de la segmentation en IP (resp. le rappel en programmes), le rappel de la segmentation en IP augmente très sensiblement. Ceci montre particulièrement bien que de nouveaux segments d'IP sont correctement inférés, produisant des résultats à peu près constants au fil du temps, alors que les résultats de segmentation par la méthode statique ont naturellement tendance à diminuer.

Enfin, nous donnons dans la figure 5.13, les résultats finaux en terme d'étiquetage, les figures de la partie supérieure représentant les résultats d'étiquetage **image par image**, et la partie inférieure en terme de **programmes**. Ces résultats sont plus mitigés en ce qui concerne l'apport de la méthode dynamique parcimonieuse. La mesure la plus pertinente de la qualité perçue par un humain de l'étiquetage est la précision en terme de programmes[3], puisqu'elle mesure effectivement le nombre de programmes faussement étiquetés. Les résultats de la méthode statique et de la méthode dynamique parcimonieuse y sont à peu près équivalents. Les résultats de la méthode dynamique parcimonieuse subissent parfois des chutes importantes, dûes à un mauvais étiquetage de bandes annonces, par la méthode présentée dans la section 5.4.2. Ceci produit une sur-segmentation, et rend donc difficile l'alignement par DTW (exemple de la journée

[3]Sur les graphes d'étiquetage en terme de programmes, la méthode dynamique n'est pas présente car les résultats par programme n'ont pas de sens vu que la segmentation est totalement erronée.

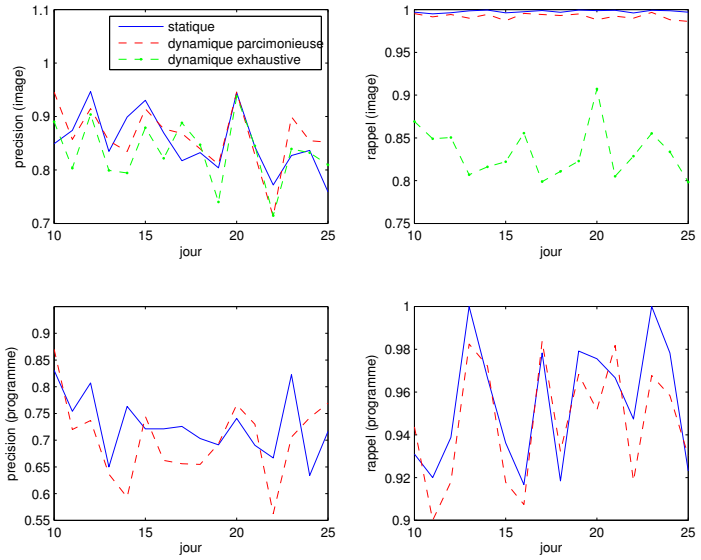

FIG. 5.13 – Comparaison des résultats d'étiquetage obtenus par les méthodes statique, dynamique exhaustive, et dynamique parcimonieuse.

22).

Nous sommes ici limités par la taille du corpus, qui ne permet pas de voir une véritable baisse de l'étiquetage par la méthode statique. L'ancienneté de l'EVR agit surtout sur la qualité de la segmentation en P/IP mais relativement peu sur la qualité de l'étiquetage.

5.5.4 Comparaison des méthodes de mise à jour

Nous étudions ici l'impact des différentes méthodes de mise à jour sur les résultats. La figure 5.14 donne les résultats de segmentation en IP dans sa partie supérieure, et les résultats d'étiquetage en terme de programme dans sa partie inférieure. Ces résultats sont donnés pour les trois types de mise à jour, définis en section 5.4.3, ainsi que pour la méthode statique, à titre de référence. Les résultats sont très proches les uns des autres, mais les différences de chaque méthode se distinguent particulièrement bien sur les résultats de segmentation en IP. La hiérarchie attendue, c'est à dire : statique $>$ type1 $>$ type2 $>$ type3, est respectée, à savoir qu'une mise a jour de type 3 possède le plus haut rappel mais aussi la plus basse précision.

Un bon compromis semble être d'utiliser une méthode de mise à jour de type 2, puisqu'elle donne des résultats intermédiaires en terme de segmentation, mais est la meilleure en ce qui concerne la précision de l'étiquetage par programme, qui est, selon nous, la meilleure mesure de la qualité perçue de l'étiquetage. Cette méthode mise à jour possède aussi un intérêt dans le cadre d'une approche semi-automatique. En effet,

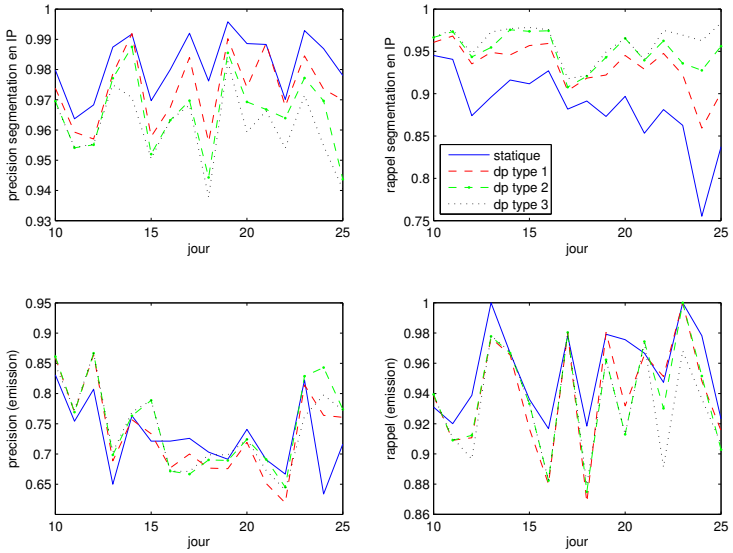

FIG. 5.14 – Comparaison des méhodes de mise à jour pour la méthode dynamique parcimonieuse.

les segments de type 2 sont assez long et facilement compréhensibles par un utilisateur (une publicité, une bande annonce...). Ces segments peuvent donc être présentés à un utilisateur pour étiquetage, ce qui est rapide et peu pénible. Il reste toutefois un problème lorsque la séquence a été mal formée : une re-segmentation manuelle est alors nécessaire, et rend la tâche beaucoup plus pénible.

5.6 Synthèse

Dans ce chapitre, une solution est apportée au problème crucial de la mise à jour de l'ensemble des vidéos de référence. Nous montrons qu'une certaine structuration des segments inférés est nécessaire, afin que la mise à jour soit pertinente. Nous proposons, à cet effet, une méthode qui permet de résoudre le problème de sur-segmentation posé par les bandes annonces, et montrons qu'au prix d'un certain différé, un étiquetage automatique de certaines séquences inférées est possible.

Les résultats montrent une nette amélioration en ce qui concerne la segmentation en programmes, et montrent que la méthode permet de réduire la dégradation des résultats au fil du temps. En revanche, en terme d'étiquetage, les résultats ne sont pas meilleurs qu'avec un ensemble de vidéo de référence statique, voir inférieurs dans certains cas. Ceci s'explique par le fait que les résultats d'étiquetage avec un EVR statique se dégradent lentement, et que notre corpus n'est pas suffisament étendu pour percevoir une réelle chute dans les résultats d'étiquetage. En conséquence, la méthode de mise à jour dynamique ne permet pas d'amélioration sur ce corpus.

Nous considérons le problème de la mise à jour comme un problème difficile, ou même avec une stratégie simple, comme celle proposée, de nombreuses difficultés apparaissent à chaque étape. Il semble nécessaire d'effectuer une analyse fine des segments inférés avant de faire la mise à jour, et cette analyse est handicapée par la variabilité du contenu, qui ne respecte pas de règles standards. Toutefois, ces premiers résultats permettent d'envisager un système de structuration fonctionnel, au moins semi-automatique, et au mieux, complètement automatique. La capacité d'auto-étiquetage est encore limitée, puisqu'elle est restreinte, pour l'instant, aux bandes annonces et aux programmes. On pourrait, toutefois, envisager d'étendre l'étiquetage à d'autres types d'inter-programmes.

Chapitre 6

Pistes de travail et perspectives

Ce chapitre propose plusieurs idées pour poursuivre le travail. Nous explorons, tout d'abord, la possibilité d'une auto-structuration d'un flux de télévision, c'est à dire sa structuration sans information extérieure, ni information a priori, à partir du seul contenu. Nous appliquons cette technique à une échelle réduite, au niveau d'un programme, avant d'envisager brièvement l'application de ce genre de technique à plus grande échelle.

Nous explorons ensuite plusieurs propositions afin d'améliorer l'étiquetage du flux. Nous explorons, dans la section 6.2, en quoi le texte présent dans la vidéo peut être utile au processus de structuration. Nous consacrons aussi une discussion à l'utilisation et la découverte de règles pour l'amélioration de l'étiquetage.

Enfin, plusieurs idées sont proposées en section 6.4, en tant que pistes de travail à explorer pour, soit améliorer la structuration, soit permettre une structuration différente.

6.1 Auto-structuration de programmes

Dans cette section, nous nous éloignons de la structuration du flux en programme, pour étudier la structuration des programmes eux-mêmes. Nous nous plaçons dans une démarche que nous appelons **auto-structuration**, car celle-ci n'utilise aucune information a priori, ni aucune information extérieure. La structuration d'un programme est réalisée à partir de son seul contenu.

Dans une deuxième partie, nous étudions le principe de l'auto-structuration à plus grande échelle, sur des flux de 24 heures, afin d'étudier les possibilités offertes par cette méthode pour la structuration de flux.

6.1.1 Introduction

La structuration de programmes est un sujet très étudié. Parmi les approches proposées, la plupart utilisent de l'information a priori. Ces approches se concentrent sur des types de programmes spécifiques (journaux télévisés, programmes sportifs), afin de réaliser une structuration de haut-niveau à l'aide d'information a priori. Nous renvoyons

le lecteur à l'état de l'art, section 1.1.2, pour plus de détails. Il existe aussi une catégorie de travaux qui tentent de ne pas utiliser d'information a priori, ou le moins possible, afin de développer une approche plus générique.

Dans le domaine de l'audio, quelques travaux se sont intéressés à extraire la structure d'un flux audio sans information a priori. L'approche de Foote *at al.* [FC03] consiste, de façon classique, à définir une distance de similarité entre trames audio, distance symétrique, et à calculer l'ensemble des distances possibles entre chaque trame, ce qui produit une matrice de similarité symétrique, de dimension N, avec N le nombre de trames du signal audio. Cette matrice de similarité laisse apparaitre des frontières, que les auteurs extraient en appliquant sur la diagonale un filtre en forme de damier, où une case pleine est représentée par une gaussienne. Lorsque deux segments bien distincts sont présents dans le signal, la matrice exhibe, en effet, une structure en forme de damier, où l'ensemble des trames appartenant à un même segment génèrent une sous-matrice carrée de fortes valeurs de similarité. Réciproquement, ces trames génèrent de basses valeurs de similarité avec les trames des autres segments, ce qui produit des zones rectangulaires. L'application du filtre en damier gaussien produit un signal monodimensionnel, dont les pics sont les frontières du document original. Cette méthode peut aussi s'appliquer à la vidéo [CF01], où l'unité de comparaison n'est plus la trame audio, mais le plan.

Concernant la vidéo, les travaux sont moins nombreux. Une approche intéressante est celle de Haidar [Hai05], déjà évoquée dans l'état de l'art, en section 1.1.3. Cette approche est générique, car indépendante du type et de la taille de la vidéo, ainsi que des desripteurs utilisés. L'idée générale consiste à considérer divers descripteurs, et à représenter leur évolution comme une série temporelle. La seule contrainte sur les descripteurs est qu'ils soient représentables comme une série temporelle mono-dimensionnelle. Haidar propose un algorithme, basé sur une recherche dichotomique, dans le but de détecter l'ensemble des couples de séquences similaires, dont la taille est comprise entre une borne minimale T_{min} et une borne maximale T_{max}. L'identification effective des couples de séquences similaires est réalisé par un algorithme calculant un *taux de couverture*, qui est une version approchée, mais rapide, de l'algorithme de recherche de la plus longue sous-séquence. Une matrice de similarité est ensuite construite à partir de cet algorithme, dont les valeurs sont les taux de couvertures pour les différentes séquences identifiées. Haidar définit aussi une mesure de similarité à partir de cette matrice, qui permet alors de mesurer la similarité entre deux documents vidéos, quelque soit leur taille. Si cette méthode possède l'avantage d'être générique, et de pouvoir traiter toutes tailles de document, son utilisation pour inférer une structure sur le document reste délicate. Une étude des techniques pour extraire une telle structure est proposée dans [Mer06]. Deux méthodes sont évaluées sur des plages de publicités, dont la méthode de Foote et Cooper, et une méthode proposée par l'auteur. Les résultats obtenus par la méthode de Merheb[Mer06] en terme de macrosegmentation sont encore faibles (47% de précision et 65% de rappel), mais prouvent qu'il peut être intéressant d'utiliser ce type de méthode.

Des méthodes plus classiques existent, qui, à partir d'une segmentation en plan d'une vidéo, et d'une définition d'une mesure de similarité entre plans, cherchent à regrouper les plans similaires. Les méthodes de clustering sont alors particulièrement intéressantes,

car elles permettent d'effectuer ce regroupement sans a priori. Odobez *et al.* [OGPG03] utilisent une méthode de clustering spectral, Zhong *et al.* [ZZC96], la méthode classique des k-means. Ce type de méthode ne crée pas, en général, de structuration dense. Les clusters formés ne représentent pas des scènes, c'est à dire un ensemble de plans temporellement proches et sémantiquement reliés, mais des plans présentant des caractéristiques visuelles similaires. La notion de similarité, et le type de vidéo analysée, doivent être selectionnés avec attention, si on souhaite que la structuration en résultant ait du sens.

Nous nous plaçons dans ce dernier cadre. Nous visons à développer une approche générique, capable de trouver une structuration sur différents types de programmes. Toutefois, tous les programmes ne se prêtent pas à une telle structuration. En particulier, les films et téléfilms ne sont pas adaptés à ce type d'approche, sauf cas particulier. Les méthodes de segmentation en scènes, c'est à dire regroupement de plans avec des contraintes temporelles, sont plus adaptées. Il existe cependant une large classe de programmes pour lesquelles les méthodes à bases de clustering ont du sens : les jeux télévisés et les émissions de plateau. Ces dernières peuvent aborder divers thèmes au cours du programme, avec plusieurs invités, et sont parfois assez longues. Il existe un intérêt important à les indexer. Notons aussi que les journaux télévisés sont aussi adaptés à l'utilisation de méthodes de clustering.

La spécificité de notre méthode vient, d'une part, que nous nous concentrons sur des données peu étudiées, les émissions de plateau, et d'autre part, que nous employons un descripteur robuste, contrairement à ce qui se fait en général. La stucturation obtenue est donc différente des résultats généralement proposés dans l'état de l'art.

6.1.2 Principe

Le principe de la méthode est tout à fait classique. Le programme est découpé en plans, et des descripteurs sont extraits pour chaque plan. On définit ensuite une mesure de similarité entre plans. Les plans sont ensuites regroupés par une méthode de clustering.

6.1.3 Distance entre plans

Nous reprenons la signature DCT, définie au chapitre 2. La signature encode une version grossière de la géométrie de l'image. Une distance entre plans basée sur ce descripteur est donc supposée caractériser la similarité des plans sur un critère purement géométrique, indépendemment de toute considération de couleur, de mouvement, ou de longueur du plan. Les plans que nous cherchons à catégoriser comme similaires partagent donc des images proches au niveau géométrique, et ne sont pas forcément contigus temporellement.

Il apparait comme une bonne idée de définir une distance robuste à l'aspect temporel, capable de trouver des similitudes entre les images des plans à différents instants de temps. Nous reprenons les distances définies dans la section 2.4.2 du chapitre 2, et nous n'en retenons que deux : la distance non-alignée (NA), pour sa simplicité, et la distance

d'édition normalisée (NED), pour ses bons taux de rappel. La distance ancrée ne peut être utilisée, car nous n'avons pas, ici, une information de position qui permet de jouer le rôle d'ancre.

Afin de réduire la complexité mémoire pour l'application de la NED, un sous-échantillonnage temporel est effectué. Cela consiste, pour un plan donné, à le sous-échantillonner uniformément, par exemple en ne conservant qu'une image sur 2, de manière à ne pas dépasser une taille de plan maximale L_{max}. Le taux d'échantillonnage est donc variable selon les plans, et est, en particulier, nul pour les plans de longueur inférieure à $Lmax$. L'échantillonnage permet de considérablement réduire les calculs, tout en résolvant le problème de la taille non bornée des plans, ce qui était problématique pour l'application de la NED.

6.1.4 Clustering

Suite à la définition d'une distance entre plans, d_P (NED ou Non-alignée), il s'agit d'organiser ces plans de façon à faire émerger une structure sur l'émission. Nous définissons le problème comme un problème de classification non supervisée, où l'on veut regrouper dans des classes les objets proches sans l'aide d'un oracle qui permettrait un apprentissage préalable.

On définit pour ceci la matrice de similarité $S = \{s_{ij}\}$ entre l'émission E_1 et l'émission E_2, comme la matrice dont l'élément s_{ij} est donné par $s_{ij} = d_P(P_i^{E_1}, P_j^{E_2})$, où $P_i^{E_k}$ est le $i^{\text{ème}}$ plan de l'émission k. Il est aisé de voir que si d_P est symétrique, ce qui est le cas pour la NED et la distance non-alignée, alors la matrice de similarité S est symétrique. Si les deux émissions sont identiques, $E_1 = E_2$, alors la matrice est carrée, et on parle alors de matrice d'auto-similarité. Á titre d'exemple, la figure 6.1 représente la matrice d'auto-similarité d'une émission de plateau de France2.

Nous proposons d'effectuer le regroupement par une méthode de clustering. Un point à mentionner est que de nombreuses méthodes de clustering font l'hypothèse que l'espace de recherche dispose d'une structure d'espace vectoriel. Ce n'est pas le cas ici, où nous ne disposons que d'un espace métrique, ce qui limite le choix des méthodes. L'utilisation d'une méthode de clustering hiérarchique semble naturelle, car elle permet de travaillller dans un espace métrique, donc directement à partir de la matrice de similarité. L'approche hiérarchique a aussi l'avantage de ne pas requérir de partitionnement préalable de l'espace ou de choix du nombre de clusters. Enfin, le regroupement en une hiérarchie peut être tout à fait intéressant car il définit naturellement une structure sur l'émission.

Odobez *et al.* [OGPG03] utilisent une méthode de clustering spectral pour structurer des vidéos personnelles et de sport. L'approche spectrale peut aussi être pertinente dans notre cas, car elle travaillle directement à partir de la matrice de similarité, et est efficace sur des données bruitées, ou sur des problèmes difficiles. Les auteurs obtiennent de meilleurs résultats qu'en utilisant des techniques classiques de k-means, ou de clustering hiérarchique. Toutefois, ces résultats ont été obtenus en utilisant des attributs peu discriminants (histogrammes de couleur), et rien ne montre que le clustering spectral soit supérieur lorsque d'autres types de descripteurs sont employés. De plus, et outre

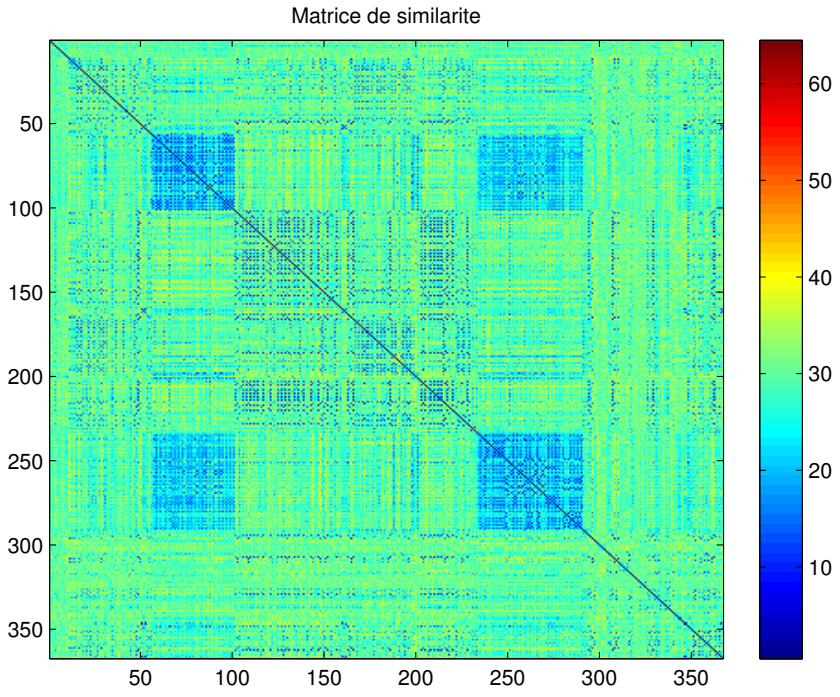

FIG. 6.1 – Matrice de similarité de l'émission « C'est au programme », émission de plateau de la chaîne France2

une complexité importante, il est délicat de choisir le nombre de clusters de manière automatique. Bien qu'intéressante, l'approche du clustering spectral n'est pas certaine de fournir de meilleurs résultats à partir d'un descripteur très discriminant comme notre descripteur DCT. Nous nous concentrons donc pour l'instant sur une méthode simple.

Il existe plusieurs variantes du clustering hiérarchique, qui sont définies par la façon dont sont regroupés les clusters à chaque itération de l'algorithme, c'est à dire la définition de la distance entre clusters. Les méthodes applicables dans notre cas sont les méthodes du lien simple, du lien complet et du lien moyen. Ce sont en effet les seules méthodes à ne pas faire l'hypothèse que la distance entre éléments est une distance euclidienne. Si l'on considère deux clusters $C_a = \{a_i\}$, $C_b = \{b_i\}$ et l'ensemble des distances entre éléments de chaque cluster $d_{ik} = d_P(a_i, b_k)$.

- La méthode du **lien simple** est la distance minimale entre deux éléments $D(C_a, C_b) = \min d_{ik}$
- La méthode du **lien complet** est la distance maximale entre deux éléments $D(C_a, C_b) = \max d_{ik}$
- La méthode du **lien moyen** est la distance moyenne entre les éléments de chaque cluster $D(C_a, C_b) = \sum_{i,k} d_{ik}$

Bien que la méthode construise une hiérarchie qui peut sembler intéressante, on

souhaite en pratique trouver des clusters qui ont un sens, et il est rare qu'un regroupement jusqu'à la racine en ait un. Il est alors nécessaire de trouver un moyen de couper la hiérarchie, afin de former des clusters qui soient homogènes du point de vue visuel. Pour cela, remarquons que les trois distances entre clusters, (lien moyen, lien simple, et lien complet), peuvent s'exprimer en tant que combinaison de distances entre plans, qui elle même est une moyenne des distance de Hamming entre signature. La distance entre clusters peut donc s'interpréter par rapport à l'intuition que l'on a de la distance entre signatures, et nous savons, d'après la section 2.4.2, page 55, quels intervalles permettent d'obtenir des résultats corrects. Une première méthode simple consiste donc à couper les branches du dendrogramme à partir d'une certaine valeur de distance α.

Cette valeur de seuil peut fluctuer selon les séquences, et les résultats de la section suivante montrent les fortes variations lorsque ce seuil est modifié. Une méthode plus robuste est de calculer le rapport des moyennes intra et inter cluster $\bar{m}^{\alpha_i}_{intra}/\bar{m}^{\alpha_i}_{inter}$ pour une plage de seuils α_i que l'on sait susceptibles d'être pertinents. On choisit alors le plus faible seuil α_i qui vérifie :

$$\frac{\bar{m}^{\alpha_i}_{intra}}{\bar{m}^{\alpha_i}_{inter}} > r$$

Une bonne valeur de r est donnée par la vérité terrain du tableau 6.1. En pratique on choisit $r = 0.4$.

GT : **liste** ; *[ensemble des clusters de la vérité terrain]*
C : **liste** ; *[ensemble des clusters à valider]*

Tant que $(GT \neq \emptyset)$ **faire**
  Pour chaque cluster GT_k de GT **faire**
    $i = Arg\max_k Card(GT_k \cap C_i)$
    $N_b = N_b + Card(GT_k \cap C_i)$
    $N_f = N_f + Card(C_i \backslash \{GT_k \cap C_i\})$
    $N_m = N_m + Card(GT_k \backslash \{GT_k \cap C_i\})$
    [update de l'ensemble des clusters]
    $C = C \backslash C_i$
  Fin Pour
  Si $(C \neq \emptyset)$ **Alors**
    $N_f = N_f + Card(C)$
  Fin Si
  Précision $= \frac{N_b}{N_b+N_f}$; Rappel $= \frac{N_b}{N_b+N_m}$
Fait

Algorithme 11: Calcul de la précision et du rappel pour la validation du clustering

6.1.5 Résultats

Nous avons sélectionné trois séquences de test.
- *JT_15_05_2005*, un journal télévisé de France2 d'une durée de 40 minutes.
- *cap_15_05_2005*, un extrait de 36 minutes de « C'est au programme », émission de plateau.
- *Orchestre*, un extrait d'1h30 d'une retransmission télévisée d'un concert symphonique.

Pour l'ensemble des résultats présentés ci-après, nous donnons les courbes précision/rappel à partir d'une vérité terrain réalisée par nos soins. Les nombres de bonnes détections N_b, de fausses détections N_f, et détections manquées N_m sont calculés par l'algorithme 11. À partir de ces valeurs, on calcule alors la précision et le rappel, par la formule 4.1. Les différentes valeurs de précision et de rappel sont obtenues en faisant varier la valeur du seuil α. Lorsque non précisée, la distance entre clusters qui est utilisée est celle du lien moyen.

La vérité terrain est très simple à réaliser pour le journal télévisé, puisqu'il suffit de regrouper ensembles les plans du présentateur, ainsi que ceux des invités. Il existe aussi une séquence d'introduction dans laquelle sont présentés les reportages. Chaque plan de l'introduction est mis en correspondance avec son quasi-double, répété durant le journal. En revanche, pour l'émission de plateau, la vérité terrain est plus subjective. Le principe général est de former un cluster pour chaque prise de vue différente du plateau. Une personne apparaissant seule à l'écran formera ainsi un cluster, mais les plans montrant deux de ces personnes en même temps formeront aussi un cluster. Aucune vérité terrain n'a été effectuée sur la séquence *Orchestre*.

Les résultats sont présentés en trois parties. La section 6.1.5.1 consiste à évaluer la dégradation introduite par l'échantillonnage temporel. La section 6.1.5.2 évalue les performances des deux distances entre plans proposées, et la section 6.1.5.3 évalue les méthodes de lien pour le clustering hiérarchique.

6.1.5.1 Effet de l'échantillonnage temporel

Nous étudions dans cette partie si l'échantillonnage a un effet sur les résultats.

Les figures 6.2 pour la NED et 6.3 pour Non-alignée donnent les courbes précision/rappel pour une valeur haute $L_{max} = 1000$, qui occasionne très peu, ou pas du tout, de ré-échantillonnage, et une valeur basse $L_{max} = 200$. Ces figures montrent que l'échantillonage n'a que peu d'effet sur les résultats, voire même une légère amélioration dans certains cas. En pratique, on choisit une valeur légèrement supérieure à la longueur moyenne d'un plan, $L_{max} = 200$, afin de réduire les plans les plus longs, tout en conservant suffisament d'information sur les plans les plus courts.

6.1.5.2 Évaluation des distances entre plans

Nous comparons les différentes distances entre elles en calculant la moyenne et la variance intra et inter-cluster, à partir de la vérité terrain. Le tableau 6.1 donne ces

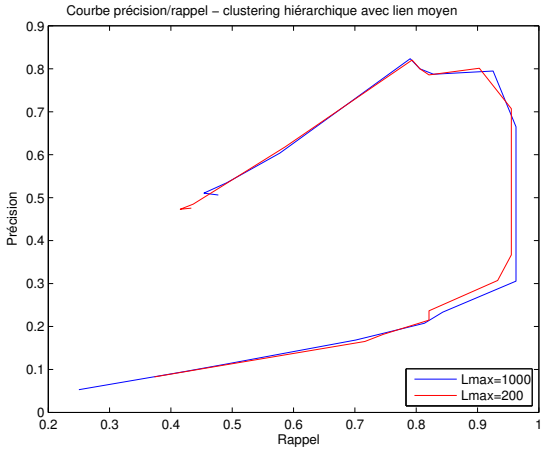

FIG. 6.2 – Effet de l'échantillonage sur la NED - séquence JT_15_05.

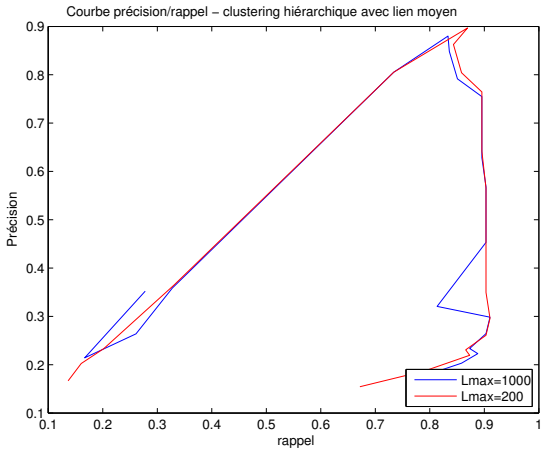

FIG. 6.3 – Effet de l'échantillonage sur la distance non-alignée - séquence JT_15_05.

résultats, exprimés en terme de distance de Hamming moyenne par signature. Les résultats en terme de TEB[1] peuvent être obtenus en divisant simplement la moyenne par la taille de la signature, soit 64. L'échantillonnage est le même pour les trois distances.

La figure 6.4 montre les performances respectives de la NED et la distance non-alignée. On peut légitimement être étonné du fait que la distance non-alignée soit parfois meilleure que la NED. En théorie, la NED cherche le meilleur alignement entre les signatures, qui permet d'obtenir le minimum de la somme des distances le long du chemin. On devrait donc toujours avoir un score de la NED supérieur à celui de la distance non-alignée. On sait d'ailleurs, dans le cas général, que la distance de Hamming est une distance d'édition particulière, et que lorsque les séquences ont même longueur,

[1] Le TEB est défini page 55

	intra-cluster		inter-cluster		intra/inter
	$\bar{m}$	σ	$\bar{m}$	σ	$\bar{m}$
NED	9.9	4.4	29.2	3.9	0.34
NA	11.7	5.9	30.9	3.8	0.38

TAB. 6.1 – Estimation des moyennes et variances intra et inter cluster à partir de la vérité terrain, pour la distance non-alignée et la NED.

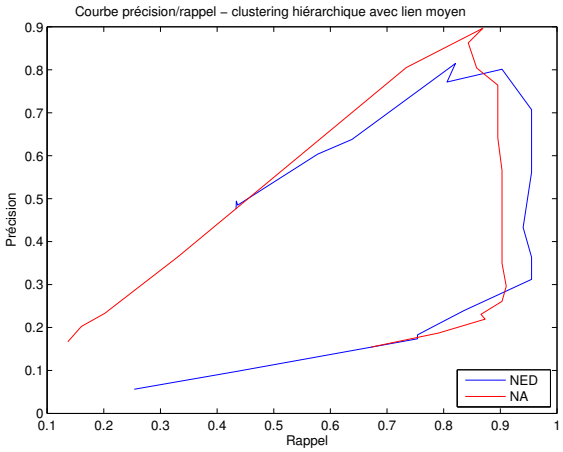

FIG. 6.4 – Courbes pour la NED et distance non-alignée - séquence JT_15_05.

la distance de Hamming est la borne supérieure de la distance d'édition. Ceci suppose toutefois que la distance locale utilisée afin de mesurer la similarité soit sans bruit. En pratique, le bruit, c'est à dire la faiblesse du descripteur, fait que des incohérences locales existent. La NED peut donc trouver des alignements non pertinents, d'où une chute dans la précision, mais un meilleur taux maximal de rappel.

6.1.5.3 Évaluation des méthodes de clustering

Une rapide comparaison des différentes méthodes de lien pour le clustering hiérarchique, donnée sur la figure 6.5, montre que la méthode du lien moyen domine les deux autres. La méthode du lien simple souffre d'un effet de chainage, c'est à dire que deux clusters peuvent être fusionnés s'ils contiennent chacun deux plans proches. La méthode du lien complet, quant à elle, forme des clusters trop compacts.

La figure 6.6 donne des résultats sur une séquence différente, la séquence *cap_15_05*. Les résultats sont beaucoup plus faibles que sur la séquence du journal télévisé. Le descripteur DCT devient ici trop discriminant, le clustering obtenu a du sens, mais il est trop précis. Ces résultats montrent la limite de la distance basée sur le descripteur DCT pour ce type de tâche, et plaident pour une étude plus complète, avec différents descripteurs, et différents contextes.

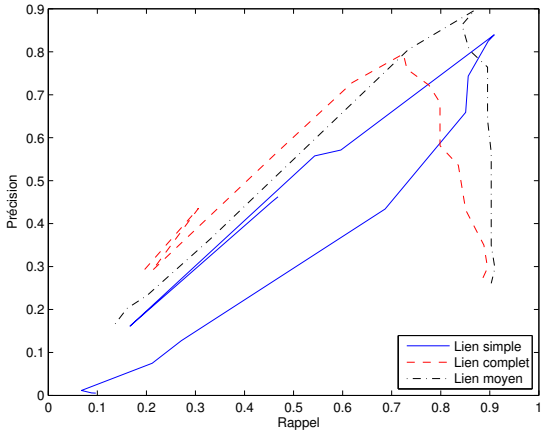

FIG. 6.5 – Comparaison des méthodes de clustering hiérarchiques - séquence JT_15_05.

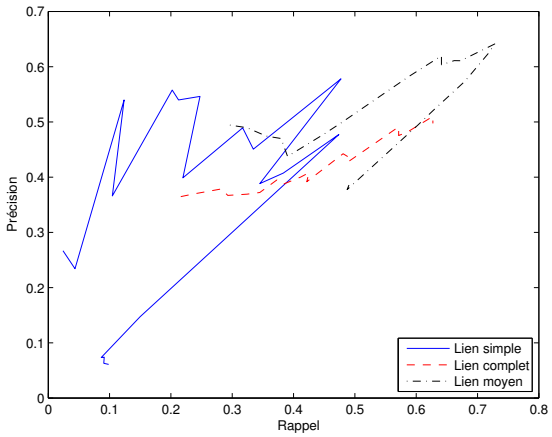

FIG. 6.6 – Comparaison des méthodes de clustering hiérarchiques - séquence cap_15_05.

6.1.6 Synthèse sur l'auto-structuration de programmes

Nous donnons sur la figure 6.7 des résultats partiels des clusters formés par la méthode, dans un cas où ceux-ci sont cohérents visuellement. Il est fréquent que les clusters soient composés de plans d'une seule et même personne. Plus généralement, les clusters sont constitués de plans qui proviennent du même angle de prise de vue, ou d'un angle très similaire. Il est donc fréquent qu'une même personne soit présente dans plusieurs clusters, en fonction des différentes prises de vues réalisées pendant l'émission. Cette contrainte, qui peut paraitre forte, a cependant un intérêt, qui est justement celui de la structuration par auto-similarité. On peut supposer que lorsque de nombreux plans d'une même personne sont réalisés avec le même angle, cet ensemble de plans parta-

geront le même sujet, par exemple une interview, et que le changement de prise de vue indiquera une rupture sémantique. C'est ce qui se passe sur l'émission « C'est au programme ».

À noter que la méthode obtient d'assez bons résultats sur le journal télévisé, ce qui n'est pas surprenant, sachant que les journaux télévisés ont une structure simple de type présentateur/reportage. Le journal contient, toutefois, une interview simultanée de deux personnes, ce qui complique légèrement la tâche de structuration, puisque l'on a alors une structure de la forme présentateur/invité1/invité2.

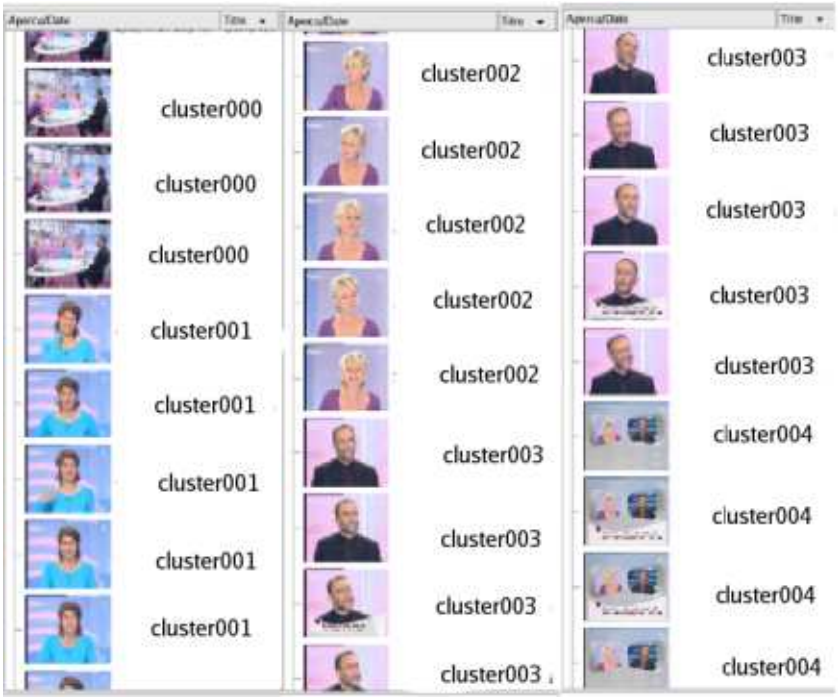

FIG. 6.7 – Exemple visuel de clusters trouvés sur l'émission « C'est au programme » de France2.

Le dernier programme sur lequel nous avons effectué des tests est une retransmission télévisé d'un concert symphonique, avec une chanteuse soliste. Il n'existe malheureusement pas de vérité terrain pour ce programme, mais il est intéressant d'étudier les clusters formés. Ceux-ci sont constitués des plans de la même prise de vue : plan large de l'orchestre, différents plans rapprochés, un cluster pour chaque gros plan sur un musicien ou un groupe de musiciens (clarinettistes, flûtistes, percussionistes, la soliste...). Les clusters formés sont donc pertinents, même s'ils sont très (trop ?) compacts.

La méthode que nous proposons construit une structure sur un programme de télévision, qui n'est pas un découpage temporel du programme en scènes, ou en thèmes, mais se rapproche plus d'une approche par *table des index*, en regroupant ensemble les plans très proches (plans du même intervenant, même prise de vue). Ce genre de méthode peut être utile pour créer un résumé de l'émission, en sélectionnant, par exemple,

les clusters les plus peuplés, et de les considérer comme caractéristiques de l'émission. Une autre application serait de pouvoir caractériser un type d'émission en fonction du nombre et de la taille des clusters par rapport au nombre de plans total de l'émission. Les émissions avec de nombreux plans similaires, par exemple les émissions de type jeu télévisé, produiraient sans doute une signature typique.

Dans une optique structuration, la structure trouvée est, à notre sens, d'encore trop bas niveau pour être réellement exploitable, et nous la pensons plus comme une première étape, par exemple pour guider des processus d'extraction d'information utilisant d'autres médias. Une extension de ce travail serait de pouvoir étiqueter les clusters trouvés, par exemple par un système de règles.

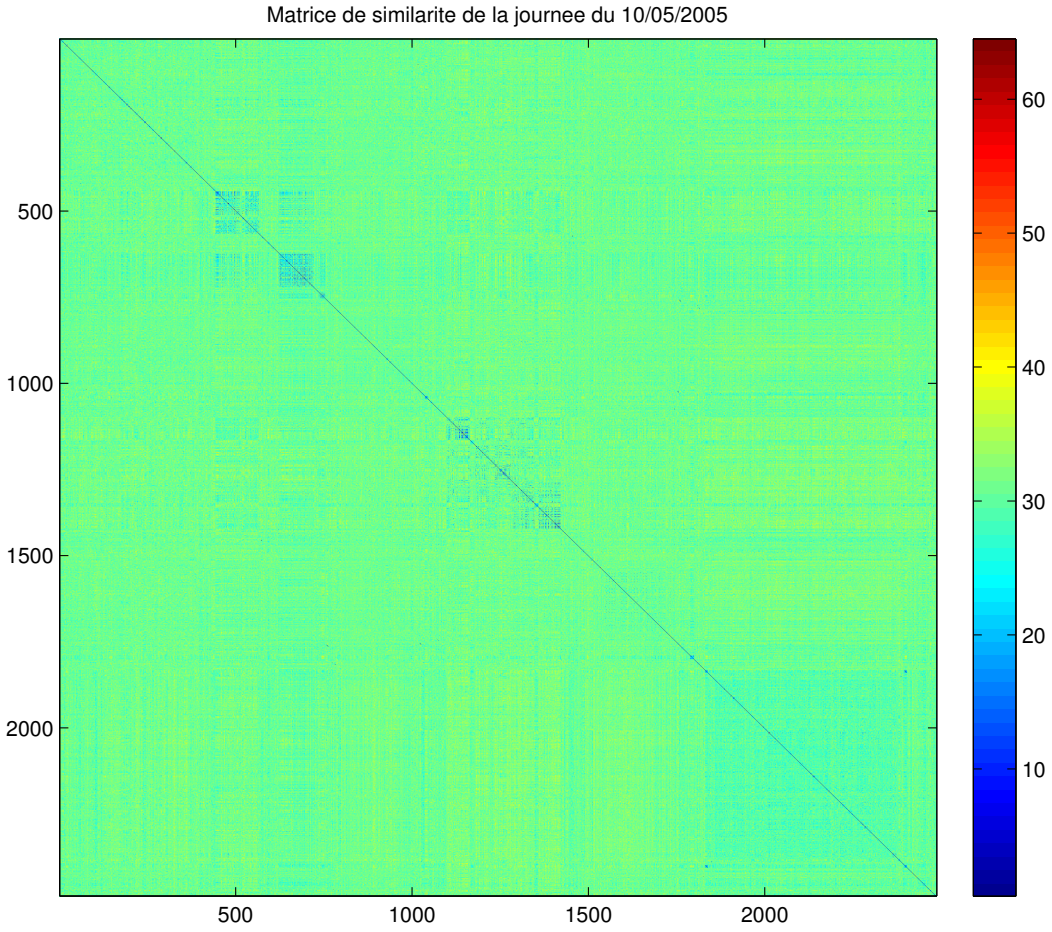

FIG. 6.8 – Matrice de similarité d'une sous-partie de la journée du 10/05 (environ 8 heures).

6.1.7 Auto-structuration à grande échelle

Dans cette section, nous nous intéressons à la possibilité d'appliquer une méthode semblable à celle présentée dans la section précédente, mais à plus grande échelle. Plus précisément, nous souhaitons calculer l'ensemble des distances entre couples de plans, à l'échelle de la journée. Sachant qu'une journée contient, en moyenne, 20.000 plans, ceci va générer une matrice de similarité de dimension 20.000, ce qui est gigantesque. Il semble difficile d'appliquer une méthode de clustering hiérarchique sur une matrice de cette taille. Nous nous contentons d'étudier visuellement le résultat, de la même manière que [Hai05].

Le nombre de calculs étant très élevé, la distance entre plans la plus adaptée à cette situation est la distance de Hamming non-alignée, qui est la plus rapide. Nous donnons des résultats visuels, en donnant directement la matrice de similarité, figure 6.8. Cette figure est la matrice d'auto-similarité, c'est à dire l'ensemble des distances entre couples de plans, sur une partie de la journée du 10/05/2005.

Il est indéniable que certaines structures apparaissent sur la matrice d'auto-similarité, notamment autour de la diagonale. Ces structures reflètent la similarité intra-émission, leur présence n'est donc pas systématique le long de la diagonale. Ce sont typiquement des émissions produites en studio, de type jeu télévisé, ou émission de plateau, qui produisent ce genre de structures.

Un point intéressant est de calculer une matrice d'inter-similarité, c'est à dire, la comparaison plan à plan de deux journées différentes. Ceci est réalisé sur les journées du 9/05 et du 10/05, et la matrice résultante est donnée par la figure 6.9. Cette matrice possède beaucoup moins de structures repérables visuellement, mais quelques structures sont tout de même présentes. il serait intéressant d'appliquer une méthode du type de Foote *et al.* [FC03], afin de déterminer si une telle structure est détectable. Toutefois, extraire de l'information pertinente d'une telle masse d'information reste un problème ouvert, et un traitement systématique sur un tel ensemble de données est certainement une piste à explorer, dans une optique de découverte de la structure du flux.

Une autre piste est de calculer une matrice de similarité au niveau audio, de la même manière que dans [FC03, Sig04]. Il serait intéressant d'étudier les différences avec la matrice de similarité image. Il se pose alors les problèmes classiques des systèmes de fusion audiovisuelle : fusion tardive ou précoce ? Haidar [Hai05] choisit une fusion précoce, ce qui semble donner des résultats satisfaisant. Toutefois, ces problèmes d'extraction et fusion d'information sur des matrices de similarité de grande taille, sont des problèmes ouverts, et méritent un travail plus approfondi.

6.2 Apports du texte pour la structuration

Cette section explore les possibilités apportées par la détection, le suivi, et la reconnaissance de texte[2]. Nous renvoyons le lecteur à Chesnel [Che06] pour un état de l'art sur les méthodes de détection et de suivi de texte.

[2] Ce travail a été effectué en collaboration avec Guillaume Chesnel, lors de son stage ingénieur [Che06]

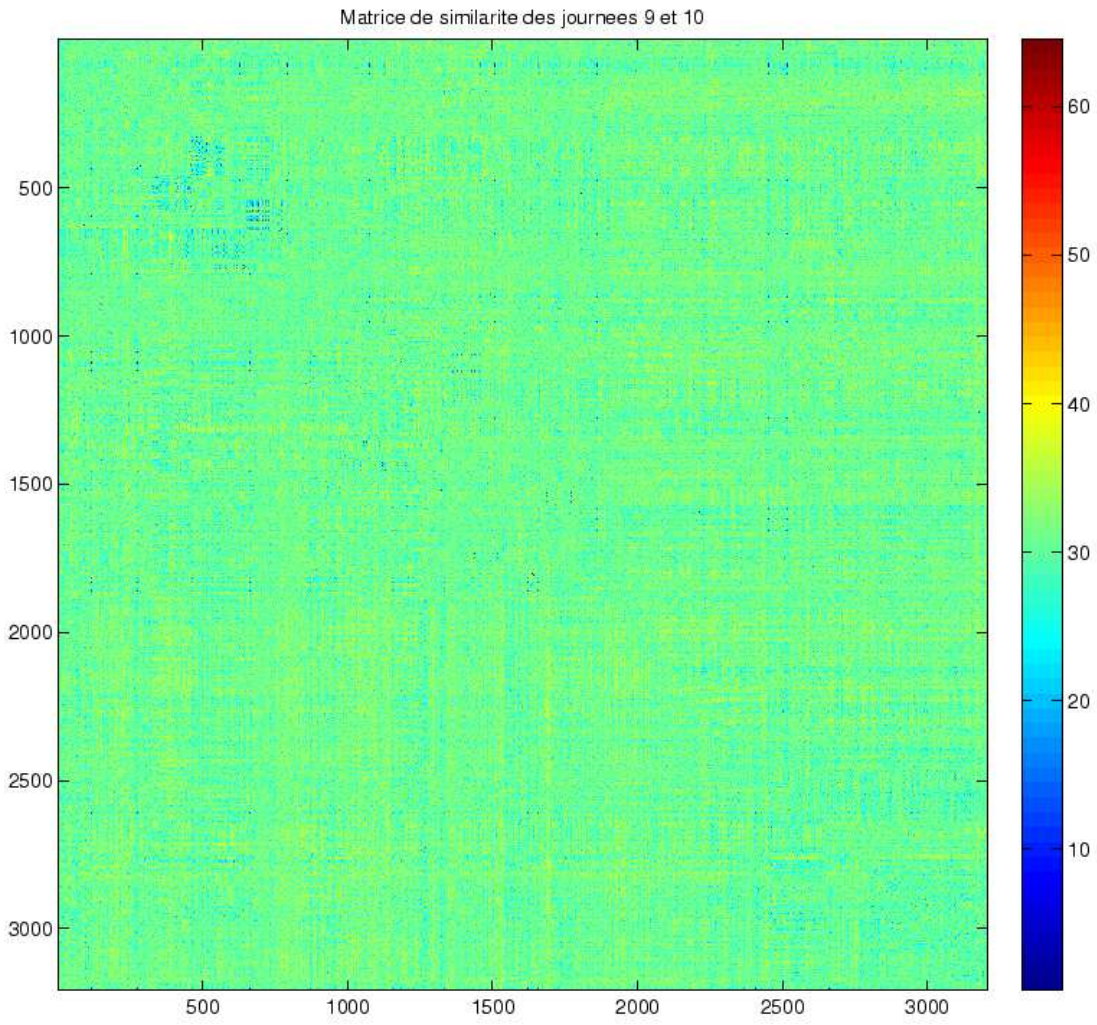

FIG. 6.9 – Matrice de similarité entre des extraits des journées du 9/05 et du 10/05 (environ 12 heures).

6.2.1 Suivi et reconnaissance de texte

La méthode de détection de texte utilisée ici a été développé lors du projet FERIA [Car04], elle est fortement basée sur les travaux de Wolf [WJ03].

La détection de texte dans des images naturelles génère énormément de fausses alarmes, il est essentiel de coupler la détection avec une méthode de suivi temporel, afin de filtrer ces fausses détections. La méthode de suivi la plus simple consiste à ne

conserver que les détections qui restent à la même position pendant un certain temps. C'est la technique utilisée dans le projet FERIA. elle ne permet pas de suivre du texte en mouvement. Par pallier cet inconvénient, nous avons développé une méthode de suivi de texte en mouvement, qui est détaillée en annexe D.

Suite à la détection, il est possible d'effectuer un traitement OCR sur les images binarisées résultats. Malheureusement, les OCR classiques ne sont pas étudiés pour gérer des images fortement bruitées, issues de la détection, puis binarisation. En conséquence, les taux de reconnaissance (lettre) sont faibles, de l'ordre de 30 à 40%. Une tentative d'amélioration des résultats par une méthode classique de N-grammes et de l'algorithme de Viterbi, d'après les travaux de Neuhoff [Neu75], a été effectuée, mais n'a pas donné d'amélioration satisfaisante des résultats. Il est aussi raisonnable de penser que le suivi permettrait d'améliorer les résultats, en prenant en compte la redondance temporelle, comme dans les travaux de Marquis et Bres [MB03].

On pourra se reporter à [Che06] pour plus de détails et de résultats sur l'ensemble du processus de détection, de suivi, et de reconnaissance.

6.2.2 Applications

6.2.2.1 Détection de génériques

La détection et le suivi de texte peuvent nous aider de plusieurs manières à améliorer nos résultats. La première application est d'améliorer la segmentation en programme, en détectant les génériques de fin. Ces derniers comportent souvent, en effet, une liste défilante des personnes impliquées dans le programme en question. La détection du générique de fin permettrait donc d'obtenir la borne de fin d'un programme avec plus de précision, ou permettre une segmentation, lorsque deux programmes ne sont pas séparés par un inter-programme (cas de certains programmes la nuit).

L'idée est de considérer qu'un générique peut se caractériser comme étant une séquence qui comporte un suivi de texte continu, d'une certaine durée minimale. Une durée minimale de suivi de 30s a été déterminée empiriquement comme pertinente. Afin de tester la détection de générique, la détection de texte a été appliquée sur deux séquences vidéo comportant des génériques.

La première séquence, d'une durée de 16 minutes, comporte les dernières minutes d'une série, le générique de fin, suivi d'un inter-programme comportant jingle et publicité. Sur cette séquence, l'ensemble du générique a été correctement détecté, et il n'y a pas eu de fausses alarmes. La figure 6.10 montre un exemple de suivi de texte sur quelques images du générique en question.

La deuxième séquence, d'une durée de 8 minutes, comporte de la même manière les derniès minutes d'un film, le générique, et des inter-programmes. Dans ce cas, seul 71% des images du générique ont été correctement identifiées, mais, comme précédemment, il n'y a pas de fausses alarmes. Autrement dit, si on considère le processus comme une classification des images en tant que générique/non-générique, le rappel est de 0.71 et la précision de 1.

La détection de texte est malheureusement un processus coûteux en temps de cal-

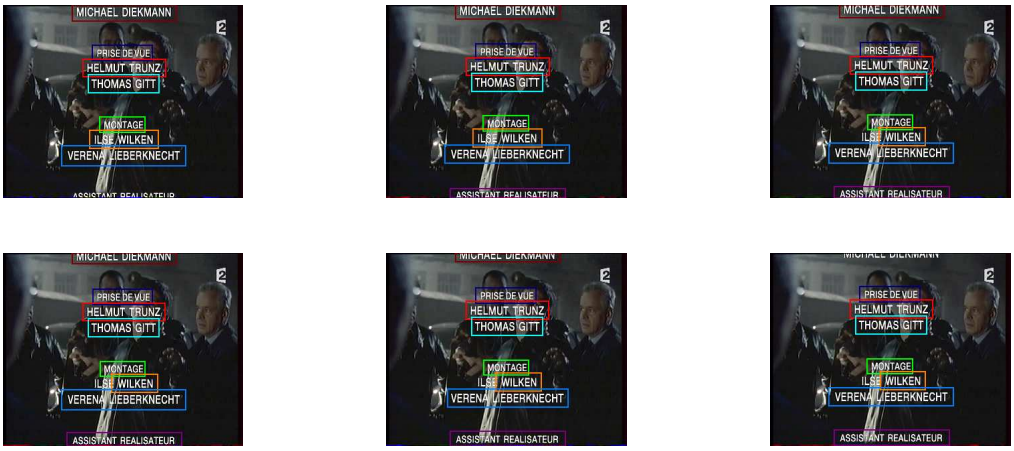

FIG. 6.10 – Exemple de suivi de texte sur un générique de série télévisée, sur 6 images consécutives.

cul, et il est difficile, pour l'instant, d'appliquer la détection de texte sur de très grands volumes de vidéo. Nous n'avons donc malheureusement pas eu le temps d'évaluer les fausses alarmes qu'une telle application générerait, et si elle permet effectivement d'améliorer la segmentation.

6.2.2.2 Étiquetage par reconnaissance de texte

La deuxième application concerne l'utilisation de la reconnaissance de texte. Cette dernière peut être très utile pour confirmer ou compléter les résultats de l'étiquetage. Les faibles résultats de la reconnaissance de texte imposent tout de même de l'utiliser avec discernement.

Un cadre intéressant d'utilisation de la reconnaissance de texte est la mise à jour de l'EVR, par la méthode dynamique parcimonieuse du chapitre 5. Dans la section 5.4.3, page 138, sont définis trois types de segments inférés, dont les segments regroupés en séquences, mais non étiquetés. Ces segments sont typiquements des publicités, des bandes-annonces, ou du parrainage, et contiennent quasiment toujours du texte. Il est intéressant d'étiqueter, même imparfaitement, ces séquences, puisque nous ne disposons d'aucune information sur celles-ci. Ces segments sont donc de bons candidats sur lesquels appliquer l'algorithme de reconnaissance de texte. En particulier, les bandes annonces sont intéressantes à étiqueter, afin d'éviter le problème de sur-segmentation, comme expliqué en section 3.4.1, page 90. De plus, une bande annonce dispose d'un avantage sur un autre type IP, c'est que le programme qu'elle annonce est présent dans le guide des programmes. La chaine de caractères à reconnaitre est donc connue. Pour chaque résultat de reconnaissance, on peut alors parcourir l'ensemble des étiquettes du guide sur plusieurs jours, afin de trouver l'étiquette qui se rapproche le plus du résultat de reconnaissance.

Là encore, l'intérêt de cette idée n'a pas pu être confirmée par des tests exhaustifs. Toutefois, des tests ont été effectués sur trois bandes annonces, en appliquant la

reconnaissance de texte sur les résultats du processus de suivi.

Boulevard du palais Bande annonce de 41 secondes d'un téléfilm. Le texte « Boulevard du palais » est présent durant toute la bande annonce, et est correctement suivi. Le meilleur résultat fourni par la reconnaissance de texte est « Boulevard _ », mais la plupart des résultats sont de très faible qualité : « h,'_l », « _WA_ DU_ ».

Tout vu, tout lu Pré-annonce de 18 secondes d'un jeu. Le texte « Tout vu tout lu » n'est présent qu'en fin de bande annonce, sur peu d'images, et n'est pas détecté. En revanche, le texte « Dans un instant » est bien suivi mais les résultats de reconnaissance sont de très faible qualité : « dansq », « _n4 rm », ou encore « _ns un ins&_ _ ».

C'est au programme Bande-annonce de 33 secondes d'une émission de plateau. Le texte « C'est au programme » n'est présent qu'en tout début et fin de bande annonce, soit environ 4 secondes, mais le texte est correctement suivi. Les résultats sont corrects, avec, par exemple « _estau _ Drogramme », « c'est au _or_ogramme ».

6.3 Utilisation et découverte de règles

6.3.1 Intérêt

Afin d'aller plus loin dans l'amélioration des résultats de l'étiquetage, nous pensons qu'il est indispensable d'introduire des connaissances sur les habitudes de diffusion de la chaîne. Plusieurs études soulignent le caractère répétitif de la structure du flux télévisé [Dom00, Pol07], dont le but est de ne pas dérouter le télespectateur, c'est à dire l'habituer à reconnaitre la structure de diffusion propre à la chaîne. Nous pensons que, dans ce contexte, l'utilisation de règles heuristiques est difficilement contournable, si l'on souhaite obtenir de très bons résultats d'étiquetage.

À titre d'exemple, nous observons, sur la chaîne France2, que les émission de météo sont toujours encadrées par un parrainage de l'entreprise D. Cet encadrement est systématique, et exclusif. Nous pouvons alors créer une règle simple, qui s'énonce ainsi : tout programme encadré par des parrainages de l'entreprise D. reçoit l'étiquette *Météo*.

En appliquant cette règle, et en comparant les résultats par rapport à une version sans règles, on constate, sur la figure 6.11, un important accroissement des résultats en ce qui concerne la mesure par programme[3]. La version avec règle montre jusqu'à 10% d'augmentation de la F-mesure, grâce à cette seule heuristique simple.

La figure montre aussi les problèmes liés à la définition d'une règle : à partir de la journée 23, les résultats avec règle chutent brutalement[4], pour devenir identiques à ceux sans règle. La raison de cette chute est simple : la chaîne a changé son habillage le

[3] La météo étant un programme court, il est normal que la mesure temporelle soit peu affectée.

[4] Les résultats sont inférieurs dès la journée du 22, puisque nos fichiers commençant en milieu de journée, le fichier de la « journée 22 » comporte, en fait, pour moitié, des programmes diffusés le 23.

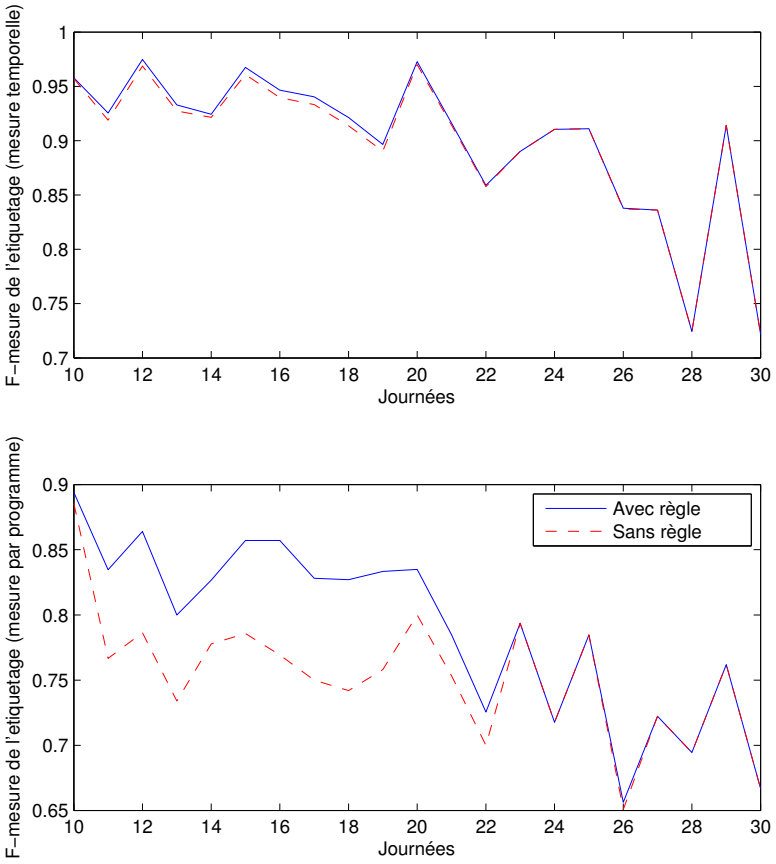

FIG. 6.11 – Influence d'une règle heuristique simple sur les résultats d'etiquetage.

23/05/2005, en raison d'un évènement sportif (Roland-Garros). La météo est toujours encadrée par un parrainage de la même entreprise, mais le contenu visuel du parrainage a changé.

La figure 6.11 montre donc, d'une part, l'importante amélioration que peuvent fournir des règles basées sur une connaissance des habitudes de diffusion de la chaîne, et d'autre part, les limites de ce genre de méthode, car elles ne sont pas génériques, et nécessitent d'être adaptées régulièrement aux changements dans la diffusion. Afin de pallier ces inconvénients, la section suivante étudie les possibilités de découverte de règles.

6.3.2 Découverte de règles

6.3.2.1 Introduction

Dans cette section un peu prospective, nous étudions la possibilité de découvrir des règles à partir des données. La section précédente a montré qu'une règle définie manuellement pouvait avoir une faible durée de vie. De plus, la définition manuelle d'une telle règle n'est pas satisfaisante, il serait intéressant de pouvoir la découvrir automatiquement.

Nous proposons d'utiliser des méthodes provenant de la fouille de données, afin de découvrir des motifs dans la diffusion des segments (de programmes ou d'inter-programmes). Les méthodes de découverte de règles d'association [HGN00, CM02], et en particulier les méthodes de découverte de suites temporelles [AS95a], nous semblent adaptées à la problématique.

La découverte de règles d'association est une technique d'extraction d'information à partir de base de données booléennes. Elle diffère sensiblement des techniques statistiques plus traditionnelles, au niveau du type de données traitées, booléennes pour les règles d'association, continues pour les statistiques, ainsi qu'au niveau du raisonnement. Les méthodes statistiques forment des hypothèses sur les données et les évaluent (choix *a priori* d'une densité de probabilité puis estimation de ses paramètres, test d'hypothèses...), alors que les méthodes de fouille de données extraient des hypothèses directement des données.

Un vocabulaire particulier a été défini pour les méthodes de découverte d'association. On considère généralement un ensemble d'éléments $I = \{i_1, i_2, \ldots, i_n\}$ appelés *items*. Un sous-ensemble de I est appelé *itemset*.

Définition 1 *Une transation, ou un enregistrement, est un vecteur binaire e, dont l'élément $e(k)$ vaut 1 si l'item i_k est présent.*

Définition 2 *Le support d'un itemset est la probabilité que tous ses items soient vrais en même temps.*

Définition 3 *Un itemset A est dit fréquent si son support est supérieur à un certain seuil, $supp(A) > supp\_min$*

Définition 4 *Une règle d'association entre deux itemsets A et B est une implication de la forme $A \rightarrow B$, où $A \cap B = \emptyset$.*

On définit deux mesures de qualité d'une règle d'association : le support et la confiance.

Définition 5 *Le support d'une règle d'association $A \rightarrow B$ est le support de l'itemset $A \cup B$,*

$$supp(A \rightarrow B) = supp(A \cup B)$$

Définition 6 *La confiance d'une règle d'association est définie par :*

$$conf(A \rightarrow B) = \frac{supp(A \cup B)}{supp(A)}$$

La base de donnée se représente, en général, comme un tableau d'éléments binaires, dont les colonnes sont l'ensemble des items, et les lignes sont les transactions.

6.3.2.2 Formalisation du problème d'étiquetage

Dans cette section, nous formalisons le problème d'étiquetage comme un problème de découverte de règles d'association. Il suffit pour cela d'identifier les bons attributs et de définir une structure identique à celle présentée dans la section précédente.

Considérons une journée de télévision représentée par un ensemble de segments, de type programme ou inter-programme, munis d'une étiquette.

- un item : une étiquette.
- un itemset : un ensemble d'étiquettes.
- un enregistrement : vecteur binaire codant la présence/absence d'une étiquette. La taille d'un enregistrement est égal au nombre d'étiquettes différentes existantes.

Grâce à ces analogies, nous pouvons utiliser la représentation de la base de données sous forme d'un tableau (ou matrice) binaire, donnée dans le tableau 6.2. Une journée est représenté par un enregistrement, qui indique, pour chaque étiquette, si elle est présente dans cette journée.

	e_1	$\ldots$	e_m	$\ldots$	e_n
j_1	a_{11}	$\ldots$		$\ldots$	a_{1n}
		$\ddots$		$\ddots$	
j_k			a_{km}		a_{kn}

TAB. 6.2 – Représentation de la base de données sous forme d'une matrice binaire. Les a_{km} sont binaires, et indiquent la présence ou l'absence de l'étiquette e_m dans la journée j_k.

Les algorithmes peuvent alors s'appliquer à ce type de données. L'algorithme *A Priori*, en particulier, qui permet d'extraire tous les k-itemsets fréquents, nous donnerait l'ensemble des segments apparaissent le plus fréquemment. La taille de l'itemset (k) est à fixer, ainsi que le support minimal exigé, pour que l'itemset soit considéré comme fréquent.

Les règles générées par un algorithme de type *A priori* ne nous sont pas très utile pour améliorer l'étiquetage, et l'algorithme n'est pas susceptible de générer des règles du type de celle définie en section 6.3.1. L'algorithme *A priori* extrait les itemsets fréquents, sans contrainte sur l'ordre des items à l'intérieur de l'itemset. Pour cela, il faut intégrer des contraintes temporelles dans la méthode. C'est l'objet de la section suivante.

6.3.2.3 Prise en compte de l'apect temporel

Les premiers travaux concernant l'extraction de règles à partir de séquences temporelles sont ceux d'Agrawal *et al.* [AS95b]. Les auteurs définissent une séquence comme étant une liste ordonnée d'itemsets, et définissent ensuite une règle d'inclusion d'une

séquence dans une autre. Le problème est alors d'extraire les séquences fréquentes maximales, c'est à dire, dont le support est maximal. Les auteurs proposent ensuite une solution basée sur des règles de filtrage des itemsets fréquents, et une modification de l'algorithme *A priori*.

Les travaux d'Höppner [Hop01] sont plus généraux, et permettent d'extraire des règles, en prenant en compte des dépendances temporelles plus complexes qu'une simple succession. Les dépendances temporelles sont exprimées par les relations d'Allen [All83]. Le même principe d'extraction de règles que dans [AS95b] est utilisé.

Ces deux méthodes peuvent s'appliquer à notre problème. Dans les deux cas, il faut définir quelles sont les relations temporelles prises en compte, ainsi que la notion d'itemset fréquent, c'est à dire, définir le seuil pour lequel on considère qu'un itemset est fréquent. Il y a donc nécessité d'avoir un a priori sur le type de motif à découvrir, et d'éventuellement, de définir plusieurs types de relations temporelles, afin de détecter indépendemment différents types de motifs. Dans le cas des travaux d'Agrawal *et al.*, ces relations temporelles sont gérées par la définition de la règle d'inclusion d'une séquence dans une autre, et dans le cas des travaux d'Höppner, par des contraintes exprimées en terme de relations d'Allen.

6.4 Autres Perspectives

6.4.1 Utilisation de la transcription de la parole

La transcription de la parole est le processus qui consiste à transformer le flux de parole, présent dans la piste audio de la vidéo, en la suite de mots prononcés, sous la forme de chaines de caractères. La transcription de la parole génère des informations de haut niveau sur le contenu d'une émission. Les faibles performances des systèmes de transcription automatique de la parole nécessitent cependant que les résultats soient utilisés avec discernement. Nous ne nous intéressons pas ici à des problèmes difficiles, comme par exemple inférer le thème d'une émission à partir de sa transcription, mais plutôt à des problèmes d'extraction de mot-clé, plus à même, selon nous, de donner des résultats intéressants dans notre contexte.

Il n'est cependant pas obligatoire de réaliser une transcription de la bande sonore pour extraire des mots-clés. Il existe des systèmes, citons par exemple Pinquier [Pin04], qui consistent à choisir et apprendre les mots-clés, pour pouvoir ensuite les reconnaitre. Cette approche est difficile ici, puisque le contenu des émissions est a priori inconnu.

Dans le contexte de cette thèse, nous envisageons deux applications principales pour la transcription. Toutes deux se situent après l'application de l'ensemble du processus d'étiquetage, et de mise à jour, sur un corpus de télévision.

La première application est d'essayer d'étiqueter des segments non étiquetés, ou dont l'étiquetage est considéré comme peu fiable, par un, ou éventuellement plusieurs, mots-clés. Nous pensons plus spécifiquement aux segments de programmes courts, pour lesquels l'étiquetage est souvent erroné. Les autres types de segments visés sont les segments inférés lors de la mise à jour, et qui n'ont pas d'étiquette. L'idée est ici d'extraire un ou des mots-clés de ces segments à partir de la transcription, par exemple par une

méthode classique de type tf-idf [SJ72], éventuellement en prenant en compte le fait que les mots issus de la transcription peuvent être erronés. On peut, ensuite, utiliser ce(s) mot(s)-clé(s) afin soit, d'étiqueter le segment, ou soit, de proposer une information complémentaire, dans le cas d'un programme à étiquetage douteux.

La deuxième application, qui n'est pas directement reliée à la structuration, est de travailler sur une collection d'émissions, par exemple, l'ensemble des jeux télévisés « Des chiffres et des lettres », diffusés sur une durée de deux semaines. Le fait de travailler sur un ensemble cohérent d'émissions permettrait, éventuellement, d'améliorer les résultats de la transcription de la parole, en choisissant un modèle de langage adapté au corpus, ce qui n'était pas possible avec l'application précédente. L'objectif serait alors d'annoter les émissions avec un certain nombre de mots-clés, qui de par leur fréquence, seront considérés comme pertinents. Dans l'exemple de l'émission « des Chiffres et des lettres », on s'attendrait à extraire des mots-clés tels que « consonne, voyelle ».

6.4.2 Utilisation de la prédiction de guide

Nous avons déjà évoqué, au chapitre 4, page 103, l'utilisation de guides prédits, par exemple par la technique de Poli [Pol07]. Nous rappelons que la prédiction de guide permet, à partir d'un apprentissage des habitudes de diffusion de la chaine, de fournir une segmentation du flux, dont la précision est bien plus grande que le guide prévisionnel. Les segments trouvés par la prédiction comportent une étiquette de genre, par exemple émission de service, mais ne portent pas de titre explicite. Une méthodologie d'utilisation du guide serait alors d'effectuer un alignement, par exemple par DTW, entre le guide prédit et le guide prévisionnel, afin d'obtenir un guide prédit étiqueté. Ce guide serait, ensuite, utilisé comme entrée du processus d'alignement du chapitre 4.

D'autres utilisation du guide prédit sont envisageables. Poli préconise d'utiliser le guide prédit comme une aide à la segmentation. La bonne précision du guide prédit permettrait de n'analyser qu'une faible partie du flux, ce qui permettrait de réduire la complexité, ou éventuellement, d'utiliser des détecteurs sophistiqués. Enfin le guide prédit pourrait fournir directement une segmentation dans certains cas délicats, comme la nuit, où la segmentation automatique échoue, à cause de l'absence d'inter-programmes. Enfin, on pourrait généraliser la méthode à la prédiction d'étiquettes, en particulier pour les émissions de service, de type *Point route*, ou *Météo*, dont les diffusions sont extrêmement régulières, et donc certainement prévisibles.

6.4.3 Détection/classification des inter-programmes par leurs répétitions

Nous avons souligné, lors de l'introduction du chapitre 2, combien les répétitions nous semblaient importantes pour la structuration. Toutefois, dans notre système, l'ensemble de vidéo de référence, utilisé pour détecter ces répétitions, est étiqueté manuellement. Chaque plan de l'EVR est étiqueté en tant que programme ou inter-programme. Pour supprimer la dépendance à un étiquetage manuel, il serait utile de pouvoir déterminer automatiquement le type, programme ou inter-programme, des plans contenus

dans l'EVR. Une idée serait de construire un système de classification des plans en fonction de la distribution de leurs répétitions dans le flux. En effet, intuitivement, il semble possible de caractériser les inter-programmes par leur fréquence de répétition. Certains segments de programmes, tels les génériques, se répètent aussi, mais sont distingables des IP, car ils se répètent à un horaire fixe. Il serait même possible de différencier les différents types d'inter-programmes en fonction de leur fréquence de répétition, et de la distribution de ces répétitions dans le flux. On sait, par exemple, que les bandes-annonces ont une durée de vie très courte, alors que les publicités ont, en général, une durée de vie plus importante.

Toutefois, il n'est pas certain que les répétitions seules pourraient permettre de classer correctement les plans en programmes/inter-programmes, et il est probable que d'autres attributs devraient être utilisés pour obtenir une classification correcte.

6.5 Synthèse

Ce chapitre a proposé diverses méthodes pour améliorer les résultats de l'étiquetage du flux télévisé. Le suivi de texte permet, dans certains cas, de détecter les génériques de fin, et donc d'améliorer la segmentation du flux. La reconnaissance de texte, quant à elle, permet d'extraire les titres des émission, ou des bandes-annonces directement à partir du flux. Il reste cependant un travail important à accomplir pour que la reconnaissance produise une information fiable, la limitation principale venant des OCR.

Une autre piste pour améliorer l'étiquetage est l'utilisation de règles, justifiée par la stabilité de la structure de la grille des programmes. Nous avons montré qu'une règle simple permettait d'accroitre sensiblement les résultats. De la même façon, ce genre de règles pourrait aussi pointer des incohérences (par exemple quatre diffusion successives du journal...).

Il n'est cependant pas satisfaisant d'avoir à définir de telles règles « manuellement », nous avons donc proposé une piste pour la découverte automatique de ces règles, à l'aide d'une méthode de découverte d'association. La modélisation du problème en tant que problème de découverte de règles d'association s'effectue assez naturellement, et il serait intéressant de voir si cette méthode permet, effectivement, l'amélioration des résultats d'étiquetage.

Nous proposons d'autres pistes : l'auto-structuration, une classification des plans en programmes/inter-programmes pour remplacer l'étiquetage manuel, l'utilisation de la transcription... Il existe donc de multiples pistes à explorer, qui méritent toutes d'être étudiées, leur efficacité a priori étant difficilement prédictible.

Conclusion

Cette thèse traite du problème de la structuration de flux de télévision. Nous avons présenté une chaîne complète de traitement, en proposant des outils de détection, une méthode de segmentation, une méthode d'étiquetage, ainsi qu'une méthode de mise à jour. Ces travaux montrent qu'il est encore difficile de pouvoir réaliser une structuration entièrement automatique de manière fiable, mais qu'une approche semi-automatique, avec une faible assistance humaine, est possible.

Dans cette conclusion, nous faisons une synthèse des travaux présentés, avant de tirer quelques conclusions sur notre travail et d'esquisser quelques perspectives à plus long terme.

Synthèse des travaux

Nos travaux se placent dans une problématique originale de structuration de flux vidéos de télévision. Cette problématique a très peu été explorée, et il est donc difficile de se comparer avec d'autres travaux du domaine. Nous nous sommes, toutefois, inspirés des travaux en détection des publicités, ainsi que des travaux en hachage perceptuel, afin de proposer deux outils : la détection des séparations et la détection des répétitions. L'outil de détection des répétitions est central à cette thèse, car une de ses idées clés est d'étudier les manières d'utiliser les répétitions pour la structuration. À ce titre, l'outil de détection des répétitions est présent dans la totalité des briques définies dans la thèse : la segmentation, l'étiquetage, et la mise à jour. L'originalité de la méthode proposée pour la détection des répétitions, est d'utiliser une méthode de hachage perceptuel. Celle-ci est capable de gérer d'importants volumes de vidéo, et d'effectuer des recherches extrêmement rapides, avec des performances en précision/rappel tout à fait bonnes, même si elle peut parfois pêcher en terme de rappel.

Les résultats des deux outils sont utilisés pour segmenter le flux en segments de programmes et d'inter-programmes. La méthode de segmentation utilise simplement le fait que les inter-programmes sont constitués de séparations, et que les inter-programmes sont, par nature, répétitifs. La segmentation est une phase délicate de la structuration, dont les résultats dépendent d'un a priori fort sur la structure des inter-programmes (la présence des séparations). Toutefois, nous avons aussi montré qu'une segmentation à l'aide des seules répétitions était possible, à condition de l'utiliser en combinaison avec une méthode efficace de mise à jour de l'EVR. La méthode peut échouer lorsque deux programmes ne sont séparés ni par un inter-programme, ni par une séparation.

Nous proposons ensuite d'étiqueter les segments de programmes, en leur attribuant un titre. Ceci est réalisé par un alignement avec le guide de programme, par une méthode de dynamic time warping. Cette méthode permet de réaliser un alignement global entre un guide de programme et la segmentation du flux, sans que la méthode soit dépendante d'un apprentissage, ou d'heuristiques. Plusieurs améliorations sont proposées afin de prendre en compte les informations externes apportées par la détection des répétitions. L'une de ces méthodes nécessite un apprentissage, mais nous estimons que cet apprentissage reste valable pour d'autres chaînes de télévision, et que la méthode ne nécessite donc pas de ré-apprentissage. L'étiquetage donne d'assez bons résultats, qui sont, toutefois, dépendants de la complétude de l'EVR, et de la qualité du guide des programmes.

Afin de pallier au problème de la mise à jour de l'EVR, pour la détection des répétitions, nous proposons une méthode de structuration dite dynamique, qui utilise un EVR mis à jour quotidiennement. le problème de la mise à jour s'avère assez délicat, notamment à cause du problème spécifique des bandes-annonces, qui nécessitent un traitement particulier. Bien que disposant d'un corpus important de trois semaines de télévision, ce corpus n'est pas suffisant pour évaluer la méthode de mise à jour dans de bonnes conditions.

Enfin, nous proposons quelques améliorations de l'étiquetage, à partir d'un outil de suivi de texte, et nous lançons quelques pistes pour inférer des règles susceptibles d'améliorer l'étiquetage grâce à une connaissance des habitudes de diffusion de la chaîne.

Conclusions et perspectives

La conclusion majeure de ce travail est qu'il est possible de structurer un flux de télévision par une méthode automatique. Le résultat n'est cependant pas de qualité suffisante pour être acceptable dans le cadre d'une solution d'archivage patrimonial, telle que réalisée à l'INA. Nous pensons qu'il est toutefois une bonne solution pour une structuration semi-automatique, où l'étape la plus problématique, l'étiquetage, serait, partiellement ou entièrement, réalisé/corrigé par un humain. Notons que, si l'étiquetage est la partie la plus difficile pour notre méthode, c'est en revanche une tâche aisée pour un humain.

Nous rappelons que la méthode proposée est initialisée par un EVR étiqueté manuellement, ce qui est une contrainte forte. Un axe de travail, en lien avec l'amélioration de la méthode de mise à jour, serait donc d'essayer de pouvoir construire cet EVR étiqueté avec une intervention humaine moindre. Une des directions envisageable est, comme proposée en 6.4.3, d'effectuer une classification basée sur les propriétés des répétitions, ou d'essayer d'extraire directement des structures à un niveau plus général, en se basant sur l'étude des matrices d'auto et d'inter-similarité.

Plus généralement, il serait aussi important de tester la méthode dans des conditions plus difficiles : sur d'autres chaînes, éventuellement sur des chaînes étrangères, sur des durées plus inmportantes. Nous rappelons, toutefois, que notre méthode fait l'hypothèse que le flux possède une structure proche de celle d'une chaîne généraliste française. Il

est nécessaire de trouver d'autres méthodes pour les chaînes sans inter-programmes, ou sans séparations. Ceci peut donc être une l'objet de travaux futurs, afin d'évaluer s'il existe des approches suffisament génériques pour être valables sur toute type de chaîne, ou si, dans l'état actuel des travaux, les méthodes doivent faire des hypothèses a priori sur la structure du flux de télévision.

Annexe A

Description du corpus

Nous utilisons essentiellement deux corpus, **corpus1** et **corpus2**. Les deux ont été enregistrés sur la chaine France2, sous la forme de fichier MPEG-2 en résolution (720x576).

Le corpus1 est constitué de deux fichiers :

- Le premier est d'une durée de 24 heures (2180727 images, 19046 plans) enregistré le 15/11/2004 ; il est appelé *video_24h*.
- Le deuxième fichier est d'une durée d'une heure (85601 images, 734 plans) enregistré le 16/11/2004 ; il est appelé *video_1h*.

Ce corpus possède une vérité terrain très précise, parfois jusqu'au niveau du plan, qui a été réalisée pour évaluer la détection des répétitions. Il est donc un peu limité en taille, de fait de la pénibilité de la réalisation de la vérité terrain. Ces deux vidéos ont été enregistrées sur la même chaine, elles possèdent donc un grand nombre de plans communs. Plus précisément, *video_1h* comporte 147 plans qui sont aussi présents dans *video_24h*. Réciproquement, *video_24h* possède 494 plans présents dans *video_1h*. Ce corpus est essentiellement utilisé dans la section 2.5.1.

Le corpus2 est constitué de 21 fichiers de 24 heures de vidéo, enregistrés en continu du 9 mai 2005 au 30 mai 2005, soit 3 semaines de télévision. Chaque fichier a une taille d'environ 40 Go. Ce corpus est utilisé pour évaluer la segmentation en programmes dans la section 3.4.2, pour tester le passage à l'échelle de la méthode de détection des répétitions en section 2.5.6, ainsi que dans le chapitre 4 et 5 pour l'évaluation de l'étiquetage. Ce corpus possède une vérité terrain plus grossière, au niveau du programme. Chaque journée de vidéo est segmentée et chaque segment est étiqueté en tant que programme ou inter-programme. Chaque programme possède un titre, de même que chaque inter-programme. Un inter-programme est, de plus, sous-catégorisé en parrainage, bande-annonce, publicité ou jingle. Les publicités ne sont pas distinguées les unes des autres, une plage de publicité est considérée comme un seul segment de publicité, et ne possède pas de titre distinctif, tous les segments de publicité sont intitulés « Plage de publicité ».

FIG. A.1 – Exemples d'inter-programme extraits de notre corpus, en partant de l'image supérieure gauche : bande-annonce, jingle, publicité, parrainage.

Annexe B

Propriétés de la DCT

Transformée de Karhunen-Loève

Définition

La transformée de Karhunen-Loève (KLT)[1] est une transformée qui permet de totalement décorréler une source markovienne d'ordre 1. C'est aussi la transformation qui permet la meilleure approximation de la source au sens de l'erreur quadratique moyenne si celle-ci est représentée sur une base de fonction orthogonales tronquée.

Nous reprenons ici la démarche de [RY90], où nous cherchons explicitement la transformation qui permet la meilleure approximation de la fonction cible, au sens de l'erreur quadratique moyenne, lorsque celle-ci est représentée seulement par quelques-uns de ses coefficients. Notons que l'approximation d'un signal par seulement quelques-uns de ses coefficients est exactement la démarche que nous adoptons en section 2.2.3 pour la construction de la signature.

Pour un vecteur x décomposé de façon classique sur une base orthogonale $\phi = [\phi_i]$, de dimension N :

$$x = \sum_{i=0}^{N-1} \alpha_i \phi_i$$

avec $\alpha_i = \langle x_i, \phi_i \rangle$. Nous cherchons à minimiser l'erreur quadratique moyenne (ou MSE pour Mean Square Error) du signal approché sur un sous-ensemble de vecteurs de ϕ :

$$MSE = E[(x - \tilde{x})^2]$$

avec le signal approché par ses seuls D premiers coefficients :

$$\tilde{x} = \sum_{i=0}^{D-1} \alpha_i \phi_i$$

alors il est relativement simple de montrer que la minimisation de la MSE aboutit pour la famille ϕ à l'expression de la KLT. Les vecteurs ϕ_i qui minimisent la MSE sont

[1] appelée aussi analyse en composantes principales

les vecteurs propres de la matrice d'autocovariance du signal x, $A = E[xx^T]$ et on a l'expression suivante :

$$A = \phi \begin{bmatrix} \mu_1 & & \\ & \ddots & \\ & & \mu_N \end{bmatrix} \phi^{-1}$$

Les μ_k étant les valeurs propres de A. Les vecteurs de base de la KLT sont donc les vecteurs propres de la matrice d'autocorrélation, ce qui conduit la KLT a être difficilement utilisable en pratique car la base dépend des données et de ce fait n'est pas pré-calculable. De plus le calcul est relativement coûteux, même si des KLT rapides ont été élaborées [Jai89].

Propriétés de la KLT

Pour un signal donné x, sa KLT est donnée par $v = \phi x$. La matrice d'autocovariance dans le domaine transformé devient :

$$E[vv^T] = \phi E[xx^T]\phi^T = \phi A \phi^{-1} \quad \text{puisque } \phi \text{ est unitaire}$$

et donc

$$E[vv^T] = \begin{bmatrix} \mu_1 & & \\ & \ddots & \\ & & \mu_N \end{bmatrix}$$

Les coefficients de la KLT sont donc décorrélés. De plus, on peut prouver que, de toutes les transformation unitaires, c'est la KLT qui permet de regrouper le plus d'énergie moyenne dans les m premiers coefficients. Si on définit la somme partielle $S_m(f)$ des variances des m premiers coefficients pour une transformation quelconque f par :

$$S_m(f) = \sum_{k=0}^{m} \sigma_k^2$$

alors pour tout $m \in [1, N]$ et pour toute transformation unitaire f on a :

$$S_m(\phi) \geq S_m(f)$$

Ces deux propriétés sont particulièrement intéressantes pour construire une signature. La propriété de regroupement de l'énergie dans les basses fréquences montre que la majorité de l'information est portée par les premiers coefficients et qu'ils sont donc de bons candidats pour construire une représentation réduite de l'image. La propriété de décorrelation indique quant à elle que chaque coefficient est potentiellement discriminant, c'est à dire porteur d'une information non redondante. Une mesure de similarité coefficient par coefficient a donc du sens.

Ces bonnes propriétés sont en revanche handicapées par le fait que la KLT est dépendante des données, puisque la base de projection est calculée à partir de la matrice d'autocovariance. On préfère donc en pratique utiliser des approximations. La meilleure approximation connue pour les signaux de type image est la transformée en cosinus discrète (DCT).

Transformée en cosinus discrète

La transformée en cosinus discrète (DCT) décompose le signal sur une base de cosinus :

$$\hat{X} = AXA^T$$

avec

$$A = [a_{ij}] \ , \ a_{ij} = \begin{cases} \frac{1}{\sqrt{2}} & \text{si } i = 0 \\ \sqrt{\frac{2}{N}} cos\left[(2j+1)\frac{\pi i}{2N}\right] & \text{sinon} \end{cases}$$

Il existe plusieurs types de DCT, Yip et Rao [RY90] en distinguent quatre types, dont le type-II, présenté ici, est le plus communément utilisé. et généralement simplement appelé « la »DCT. On peut aussi donner son expression développée sur un signal 2D. Pour une image I de taille (N, M) , le coefficient $DCT(u, v)$ est donné par :

$$DCT(u,v) = \alpha(u,v) \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} cos\left[(2x+1)\frac{\pi u}{2M}\right] cos\left[(2y+1)\frac{\pi v}{2M}\right] I(x,y)$$

avec $\alpha(u,v) = \frac{2}{\sqrt{NM}} C(u)C(v)$ et $C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{pour } u = 0 \\ 1 & \text{sinon} \end{cases}$

La DCT est très populaire en traitement d'images parce qu'elle approche de très près les propriétés de la KLT. Pour peu que la source respecte effectivement les propriétés d'une source markovienne d'ordre 1 et soit très corrélée (ce qui est le cas de la plupart des images naturelles), la DCT possède des performances quasi-comparables à la KLT en terme de répartition de l'énergie et décorrelation des coefficients [RY90]. Une très bonne approximation du signal est donc obtenue en conservant seulement les coefficients basses fréquences, ce qui est une propriété évidemment très utilisée en compression, mais peut être utilisé aussi pour l'identification de contenus, à condition que la déformation du contenu ne soit pas trop sévère.

De plus, la DCT est reliée à la transformée de Fourier discrète, et peut donc se calculer par des algorithmes rapides de style FFT. Nous utilisons l'implémentation de FFTW [FJ05], qui possède le double avantage d'être en licence GPL, et d'être l'une des plus rapide implémentations existante[2].

[2]Comme le montre la page web : http ://www.fftw.org/benchfft/

Annexe C

Estimation par maximum de vraisemblance

Supposons disposer d'un ensemble d'échantillons $D = (x_1 \dots x_n)$, indépendants et identiquements distribués (i.i.d) suivant une distribution $p(x|\theta)$, θ étant l'ensemble de paramètres qui régissent la distribution. L'estimation par maximum de vraisemblance consiste à calculer les paramètres représentés par θ qui maximisent la vraisemblance :

$$\hat{\theta}_{MV} = \arg\max_{\theta} p(D|\theta)$$

Pour des raisons calculatoires, on utilise souvent la log-vraisemblance, ce qui ne change rien au résultat, la fonction ln étant monotone croissante. Les échantillons étant i.i.d., la vraisemblance peut s'écrire :

$$p(D|\theta) = \prod_{i=1}^{n} p(x_i|\theta)$$

d'où la log-vraisemblance :

$$\ln p(D|\theta) = \sum_{i=1}^{n} \ln p(x_i|\theta)$$

Dans le cas d'une distribution gaussienne $p \sim N(\mu, \sigma^2)$, la loi est caractérisée par ses seules moyenne et variance. L'ensemble des paramètres θ est alors $\theta = (\mu, \sigma)$ et on peut dériver une expression analytique pour le maximum de vraisemblance. Si les échantillons suivent une loi gaussienne :

$$p(x_i|\theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp - \frac{(x_i - \mu)^2}{2\sigma^2}$$

la vraisemblance s'écrit :

$$\ln p(D|\theta) = -n \ln \sqrt{2\pi}\sigma - \sum_{i=1}^{n} \frac{(x_i - \mu)^2}{2\sigma^2}$$

On cherche alors l'extrema de $\ln p(D|\theta)$ en posant le système suivant :

$$\begin{cases} \dfrac{\partial \ln p(D|\theta)}{\partial \mu} = 0 \\ \dfrac{\partial \ln p(D|\theta)}{\partial \sigma} = 0 \end{cases}$$

En résolvant ce système, on obtient :

$$\begin{cases} \hat{\mu} = \dfrac{1}{n} \displaystyle\sum_{i=1}^{n} x_i \\ \hat{\sigma}^2 = \dfrac{1}{n} \displaystyle\sum_{i=1}^{n} (x_i - \hat{\mu})^2 \end{cases}$$

Ce résultat est très intuitif puisque l'estimation par maximum de vraisemblance d'une loi gaussienne est faite simplement en calculant la moyenne et la variance des échantillons. À noter toutefois que l'estimation de la variance est biaisée puisque l'on a $E[\hat{\sigma}^2] = \frac{n-1}{n}\sigma^2$.

On pourrait donc proposer d'autres estimateurs de la variance qui ne seraient pas biasés, tel $\dfrac{1}{n-1}\displaystyle\sum_{i=1}^{n}(x_i - \hat{\mu})^2$, mais en pratique les performances de l'estimateur par maximum de vraisemblance sont très bonnes, notamment du au fait qu'il soit sans biais asymptotiquement, i.e. $\lim\limits_{n\to\infty} E[\hat{\sigma}^2] - \sigma^2 = 0$

Annexe D

Méthode de suivi de texte

Cette annexe détaille la méthode de suivi de texte développée. Le suivi se déroule hors ligne, une fois l'étape de détection réalisée. Nous nous contentons de rappeler que le résultat de la détection de texte est, pour chaque image, une liste de *boites englobantes*, autrement dit, la détection de texte renvoie les coordonnées des zones considérées comme du texte

Nous reprenons l'idée générale de Pitié *et al.* [PBKD05], qui consiste à modéliser le problème de suivi comme une estimation de la probabilité a posteriori d'une séquence $(x_1, \ldots, x_N)$, où x est une variable aléatoire représentant la position de l'objet, et N le nombre d'images dans la séquence. Par le théorème de Bayes, la probabilité a posteriori peut s'écrire comme :

$$p(x_1, \ldots, x_N | y_1, \ldots, y_N) = p(x_1, \ldots, x_N) \prod_{k=1}^{N} p(y_k | x_k) \tag{D.1}$$

Le premier terme, $p(x_1, \ldots, x_N)$, est la distribution a priori, qui, si on considère le processus comme Markovien, est donné par les probabilités de transition entre états, c'est à dire, la probabilité de transition de la boite source vers la boite cible.

$$p(x_1, \ldots, x_N) = p(x_1) \prod_{n=1}^{N-1} p(x_{n+1} | x_n)$$

Ces probabilités de transition sont utilisées pour encapsuler les règles quant aux mouvements possibles des boites englobantes. Les zones de texte à l'écran ont, en général, des mouvements peu marqués et réguliers, afin de permettre leur lecture par le téléspectateur. Plutôt que des règles sur les mouvements acceptables, la probabilité de transition est vue comme une distance composée de deux termes : un terme mesurant la distance entre les positions des boites, et un terme mesurant leur similarité visuelle.

La similarité entre boites est définie comme la distance de Levenshtein entre les profils horizontaux et verticaux du contenu binarisé des boites englobantes. La similarité entre la boite a et la boite b est notée sim_a^b. La distance entre les positions des boites est une différence entre coordonnées des boites englobantes. Si on note x_n^{sg} la coordonnée

supérieure gauche de la boite englobante considérée à l'état x_n, et x_n^{id} la coordonnée inférieure droite, alors la distance entre position des boites est donnée par :

$$pos_n^{n+1} = |x_{n+1}^{sg} - x_n^{sg}| + |x_{n+1}^{id} - x_n^{id}|$$

La probabilité de transition est alors donnée par :

$$p(x_{n+1}|x_n) = \frac{1}{\alpha} sim_n^{n+1} pos_n^{n+1}$$

où α est un facteur de normalisation. .

Le deuxième terme de l'équation D.1 est la vraisemblance de x, étant donné le modèle y, qui correspond ici à l'image. La vraisemblance est considérée comme indépendante pour chaque image. Pitié *et al.* suggèrent de déterminer les candidats à partir des pics de cette vraisemblance. Dans notre cas, les candidats sont donnés par le processus de détection de texte, la vraisemblance est alors une mesure de confiance donnée par l'algorithme, pour chaque boite englobante détectée.

Grâce à cette modélisation, on peut alors appliquer l'algorithme de Viterbi pour trouver le meilleur chemin. L'algorithme de Viterbi doit être appliqué autant de fois qu'il existe de zones de texte à suivre. Concrètement, Viterbi est appliqué sur la première détection de la première image de la séquence, ce qui nous donne la trajectoire de cette détection dans l'ensemble de la séquence. Les candidats correspondants à cette trajectoire sont supprimés, et Viterbi est appliqué de nouveau, sur une autre détection de la première image. Le processus est répété jusqu'à ce que l'ensemble des candidats soit épuisé. Des règles sont utilisées pour gérer la disparition de l'écran des zones de texte.

Annexe E

Résultats de structuration

Cette annexe présente les résultats de structuration obtenus sur les journées du 10, 11 et 12/05/2005, du corpus 2.

Ces résultats sont obtenus par une structuration statique, avec une valeur de seuil de segmentation fixée à 1200 images. Ces exemples de structuration automatiques sont donnés ici à titre indicatif, afin d'illustrer d'une façon plus intuitive le résultat de la structuration, et de donner une idée de résultats typiques.

Verite terrain			Structuration automatique		
15h28	15h42	Le renard	15h28	15h42	Le renard
15h48	16h36	Rex	15h48	16h36	Rex
16h36	16h37	Un livre	16h36	16h37	Un livre
16h43	16h57	Des chiffres et des lettres	16h43	17h10	Des chiffres et des lettres
16h57	17h10	Des chiffres et des lettres			
17h15	17h43	Tout vu, tout lu	17h15	17h43	Tout vu, tout lu
17h44	17h51	Tout vu, tout lu	17h44	17h51	Tout vu, tout lu
17h55	18h38	Urgences	17h55	18h38	Urgences
18h42	18h44	Cd'aujourd'hui	18h42	18h44	Cd'aujourd'hui
18h49	19h39	On a tout essaye	18h49	19h39	On a tout essaye
19h41	19h42	Un jour un arbre	19h41	19h42	Un jour un arbre
			19h42	19h43	Une journee de houf
19h46	19h48	MPGMC	19h46	19h48	MPGMC
19h48	19h48	Pour Florence et Hussein			
19h49	19h55	Une journee de houf	19h49	19h55	Une journee de houf
19h55	19h56	Conso mag			
19h56	19h58	La meteo	19h56	19h58	La meteo
19h58	20h41	Journal	19h58	20h41	Journal
20h44	20h45	A parts egales	20h44	20h45	A parts egales
20h45	20h47	La meteo	20h45	20h48	La meteo
			20h48	20h49	Journal
			20h50	20h51	Journal
20h51	20h52	Plus jamais comme ca	20h51	20h52	Plus jamais comme ca
20h58	22h31	Big mamma	20h58	22h31	Big mamma
22h39	22h44	Comme au cinema l'hebdo	22h39	22h44	Big mamma
22h48	0h21	Une viree en enfer	22h48	0h21	Une viree en enfer
0h29	0h52	Journal de la nuit	0h29	0h52	Journal de la nuit
0h53	0h56	La meteo	0h53	0h56	La meteo
1h00	1h02	Cd'aujourd'hui	1h0	1h2	Cd'aujourd'hui
1h03	2h01	Histoires courtes	1h03	1h41	Histoires courtes
			1h41	2h01	Chanter la vie
2h03	2h53	Chanter la vie	2h03	2h53	Chanter la vie
2h54	3h24	Trente millions d'amis	2h53	4h23	Trente millions d'amis
3h24	3h59	Les arts en liberte			
3h59	4h23	Journal de la nuit			
4h23	4h26	La meteo	4h23	4h26	La meteo
4h26	5h51	Faites entrer l'accuse	4h26	5h51	Faites entrer l'accuse
5h51	5h52	Un livre	5h51	5h52	Un livre
5h54	6h19	Les z'amours	5h54	6h19	Les z'amours

TAB. E.1 – Resultats de structuration sur la journee du 10/05/2005. Premiere partie

Verite terrain			Structuration automatique		
6h19	6h25	Les z'amours	6h19	6h25	Les z'amours
6h25	6h26	Point route	6h25	6h26	Point route
6h29	6h55	Telematin	6h29	6h55	Telematin
6h55	6h56	La meteo	6h55	6h56	Telematin
6h56	6h57	Point route	6h56	6h57	Point route
7h00	7h28	Telematin	7h00	7h27	Telematin
			7h27	7h28	Telematin
7h32	7h51	Telematin	7h32	7h51	Telematin
7h54	7h59	Telematin	7h54	7h59	Telematin
7h59	8h00	La meteo			
8h00	8h11	Telematin	8h00	8h11	Telematin
			8h12	8h13	Telematin
8h14	8h23	Telematin	8h14	8h23	Telematin
8h24	8h25	La meteo	8h24	8h25	Telematin
8h25	8h27	Telematin	8h25	8h27	Telematin
8h27	8h28	Point route	8h27	8h28	Point route
8h29	8h30	Un livre	8h29	8h30	Un livre
8h33	8h55	Des jours et des vies	8h33	8h55	Des jours et des vies
8h59	9h20	Amour, gloire et beaute	8h59	9h20	Amour, gloire et beaute
9h21	9h23	Cd'aujourd'hui	9h21	9h23	Cd'aujourd'hui
9h26	9h52	Top of the pops	9h26	9h52	Top of the pops
9h57	10h18	KD2A	9h57	9h59	Top of the pops
			9h59	10h18	KD2A
10h22	10h46	KD2A	10h22	10h46	KD2A
10h49	10h53	Journal	10h49	10h53	Journal
10h57	11h20	Motus	10h57	11h28	Motus
11h21	11h28	Motus			
11h33	11h59	Les z'amours	11h33	12h04	Les z'amours
11h59	12h03	Les z'amours			
12h04	12h06	Cd'aujourd'hui	12h04	12h06	Cd'aujourd'hui
12h11	12h29	La cible	12h11	12h47	La cible
12h30	12h47	La cible			
12h50	12h51	Le millionnaire	12h49	12h51	Le millionnaire
12h55	12h56	Pour Florence et Hussein	12h55	12h56	Pour Florence et Hussein
12h56	12h57	La meteo	12h56	12h57	La cible
12h58	13h42	Journal	12h58	13h42	Journal
13h46	13h49	La meteo	13h46	13h49	Journal
13h50	13h51	A parts egales	13h50	13h51	A parts egales
13h53	14h41	Inspecteur Derrick	13h53	14h41	Inspecteur Derrick

TAB. E.2 – Résultats de structuration sur la journée du 10/05/2005. Deuxième partie

Verite terrain			Structuration automatique		
15h28	15h39	Le renard	15h28	15h39	Le renard
15h45	16h30	Rex	15h45	16h30	Rex
			16h30	16h31	Un livre
16h31	16h32	Un livre	16h31	16h32	Un livre
16h38	16h52	Des chiffres et des lettres	16h38	16h52	Des chiffres et des lettres
16h53	17h05	Des chiffres et des lettres	16h52	17h05	Des chiffres et des lettres
17h12	17h40	Tout vu, tout lu	17h12	17h48	Tout vu, tout lu
17h40	17h48	Tout vu, tout lu			
17h52	18h35	Urgences	17h52	18h35	Urgences
18h38	18h40	Cd'aujourd'hui	18h38	18h40	Cd'aujourd'hui
18h46	19h38	On a tout essaye	18h46	19h38	On a tout essaye
			19h39	19h40	Une journee de houf
19h40	19h41	Un jour un arbre	19h40	19h41	Un jour un arbre
19h45	19h47	MPGMC	19h46	19h47	MPGMC
19h48	19h54	Une journee de houf	19h48	19h54	Une journee de houf
19h55	19h55	Conso mag			
19h55	19h58	La meteo	19h55	19h58	Une journee de houf
19h58	20h41	Journal	19h58	20h41	Journal
			20h41	20h42	Loto
20h43	20h45	La meteo	20h43	20h45	Loto
20h48	20h52	Loto	20h48	20h52	Loto
20h52	20h53	Plus jamais comme ca	20h52	20h53	Plus jamais comme ca
20h58	22h23	Une famille pas comme les autres	20h58	22h23	Une famille pas comme les autres
22h35	0h43	Ca se discute	22h35	0h43	Ca se discute
0h50	0h56	Talents Cannes 2005	0h50	1h13	Talents Cannes 2005
0h56	1h13	Journal de la nuit			
1h13	1h16	La meteo	1h13	1h16	Journal de la nuit
1h21	1h23	Cd'aujourd'hui	1h21	1h24	Cd'aujourd'hui
1h24	3h03	Des mots de minuit	1h24	4h17	Des mots de minuit
3h03	3h48	La source de vie			
3h48	4h17	Emissions religieuses			
4h18	4h35	Journal de la nuit	4h18	4h35	Journal
4h35	4h37	La meteo	4h35	4h37	Journal
4h38	5h25	Lisbonne la bleue	4h38	5h25	Lisbonne la bleue
5h25	5h51	Outremers	5h26	5h41	Journal
			5h41	5h50	Un livre
5h51	5h52	Un livre	5h51	5h52	Un livre
5h54	6h20	Les z'amours	5h54	6h25	Les z'amours
6h21	6h25	Les z'amours			

TAB. E.3 – Resultats de structuration sur la journee du 11/05/2005. Premiere partie

Verite terrain			Structuration automatique		
6h25	6h26	Point route	6h25	6h26	Point route
6h29	6h54	Telematin	6h29	6h54	Telematin
6h54	6h56	Telematin	6h54	6h56	Telematin
6h56	6h57	Point route	6h56	6h57	Point route
6h59	7h25	Telematin	6h59	7h25	Telematin
7h25	7h27	Telematin	7h25	7h27	Telematin
7h27	7h27	Telematin			
7h31	7h55	Telematin	7h31	7h55	Telematin
7h58	8h12	Telematin	7h59	8h12	Telematin
8h15	8h30	Telematin	8h15	8h30	Telematin
8h30	8h31	Telematin	8h30	8h31	Telematin
8h31	8h37	Telematin	8h31	8h37	Telematin
8h41	8h42	Point route	8h41	8h42	Point route
8h42	8h43	Un livre	8h42	8h43	Un livre
8h44	9h02	Des jours et des vies	8h43	9h02	Des jours et des vies
			9h02	9h03	Des jours et des vies
			9h07	9h23	Amour, gloire et beaute
9h07	9h28	Amour, gloire et beaute	9h23	9h29	Amour, gloire et beaute
9h30	9h32	Cd'aujourd'hui	9h30	9h32	Cd'aujourd'hui
9h36	10h09	C'est au programme	9h36	10h09	C'est au programme
10h13	10h45	C'est au programme	10h13	10h40	C'est au programme
			10h40	10h43	Journal
			10h43	10h45	Journal
10h49	10h53	Journal	10h49	10h53	Journal
10h58	11h21	Motus	10h58	11h29	Motus
11h22	11h29	Motus			
11h33	11h59	Les z'amours	11h33	12h05	Les z'amours
11h59	12h05	Les z'amours			
12h06	12h8	Cd'aujourd'hui	12h05	12h08	Cd'aujourd'hui
12h12	12h30	La cible	12h12	12h46	La cible
12h31	12h46	La cible			
12h49	12h50	Loto	12h49	12h50	Loto
12h50	12h51	Le millionnaire	12h50	12h52	Le millionnaire
12h51	12h52	Pour Florence et Hussein			
12h55	12h57	La meteo	12h55	12h57	Loto
12h58	13h41	Journal	12h58	13h41	Journal
13h45	13h47	La meteo	13h45	13h47	Journal
13h48	13h49	A parts egales	13h48	13h49	Journal
13h51	14h38	Inspecteur Derrick	13h51	14h38	Inspecteur Derrick

TAB. E.4 – Resultats de structuration sur la journee du 11/05/2005. Deuxieme partie

Verite terrain			Structuration automatique		
15h28	15h44	Le renard	15h28	15h45	Le renard
15h50	16h36	Rex	15h50	16h36	Rex
			16h36	16h37	Des chiffres et des lettres
16h37	16h38	Un livre	16h37	16h38	Un livre
16h43	16h57	Des chiffres et des lettres	16h43	17h10	Des chiffres et des lettres
16h57	17h10	Des chiffres et des lettres			
			17h10	17h10	Des chiffres et des lettres
17h15	17h45	Tout vu, tout lu	17h15	17h51	Tout vu, tout lu
17h45	17h51	Tout vu, tout lu			
17h55	18h37	Urgences	17h55	18h37	Urgences
18h40	18h42	Cd'aujourd'hui	18h39	18h42	Cd'aujourd'hui
18h48	19h38	On a tout essaye	18h48	19h38	On a tout essaye
19h40	19h41	Un jour un arbre	19h40	19h41	Un jour un arbre
19h46	19h47	MPGMC	19h46	19h47	MPGMC
19h49	19h55	Une journee de houf	19h49	19h55	Une journee de houf
19h55	19h58	La meteo	19h55	19h58	Une journee de houf
19h58	20h32	Journal	19h58	20h52	Journal
20h32	20h52	Question ouverte			
			20h54	20h55	Question ouverte
20h55	20h57	La meteo	20h55	20h57	Question ouverte
21h01	21h03	Point route	21h01	21h03	Point route
21h04	21h04	Plus jamais comme ca	21h04	21h04	Plus jamais comme ca
21h09	23h19	Envoye special	21h09	23h19	Envoye special
23h31	1h47	Trafic.musique	23h31	1h47	Trafic.musique
1h52	1h58	Talents Cannes 2005	1h52	1h58	Talents Cannes 2005
1h58	2h00	La meteo	1h58	2h00	Journal de la nuit
2h06	2h08	Cd'aujourd'hui	2h05	2h59	Six pieds sous terre
2h08	3h00	Six pieds sous terre	2h59	3h00	Urgences
3h01	4h10	Contre-courant	3h01	4h10	Contre-courant
4h10	5h43	Contre-courant	4h10	4h45	Contre-courant
5h43	5h50	Azimuts	4h46	5h50	Contre-courant
5h50	5h51	Un livre	5h50	5h51	Un livre
5h53	6h24	Les z'amours	5h53	6h26	Les z'amours
6h25	6h26	Point route			
6h29	6h55	Telematin	6h29	6h55	Telematin
6h55	6h56	Telematin	6h55	6h56	Telematin
6h57	6h58	Point route	6h57	6h58	Point route

TAB. E.5 – Resultats de structuration sur la journee du 12/05/2005. Premiere partie

Verite terrain			Structuration automatique		
7h0	7h25	Telematin	7h0	7h25	Telematin
7h25	7h26	Telematin	7h25	7h26	Telematin
7h26	7h26	Telematin			
7h30	7h55	Telematin	7h30	7h55	Telematin
7h59	8h00	Telematin	7h59	8h00	Telematin
8h00	8h12	Telematin	8h00	8h12	Telematin
8h15	8h29	Telematin	8h15	8h29	Telematin
8h29	8h30	Telematin	8h29	8h30	Telematin
8h30	8h36	Telematin	8h30	8h36	Telematin
8h39	8h40	Point route	8h39	8h40	Point route
8h40	8h41	Un livre	8h40	8h41	Un livre
8h42	9h01	Des jours et des vies	8h43	9h00	Des jours et des vies
9h6	9h27	Amour, gloire et beaute	9h6	9h30	Amour, gloire et beaute
9h28	9h30	Cd'aujourd'hui			
9h33	10h07	C'est au programme	9h33	10h08	C'est au programme
10h11	10h38	C'est au programme	10h11	10h38	C'est au programme
10h38	10h40	La meteo	10h38	10h40	Journal
10h40	10h43	C'est au programme	10h40	10h43	Journal
10h48	10h52	Journal	10h47	10h52	Journal
10h55	11h18	Motus	10h55	11h26	Motus
11h19	11h26	Motus			
11h32	11h56	Les z'amours	11h32	12h03	Les z'amours
11h57	12h03	Les z'amours			
12h04	12h06	Cd'aujourd'hui	12h04	12h06	Cd'aujourd'hui
			12h09	12h10	Cd'aujourd'hui
12h10	12h28	La cible	12h10	12h47	La cible
12h28	12h47	La cible			
			12h47	12h49	La cible
12h50	12h51	Le millionnaire	12h50	12h51	Le millionnaire
12h55	12h57	La meteo	12h55	12h57	La cible
12h58	13h41	Journal	12h57	13h42	Journal
			13h45	13h46	Journal
13h46	13h48	La meteo	13h46	13h48	Journal
13h49	13h50	A parts egales	13h49	13h53	Journal
13h51	13h53	Point route			
13h54	14h39	Inspecteur Derrick	13h54	14h39	Inspecteur Derrick
14h45	14h46	Point route	14h45	14h46	Point route

TAB. E.6 – Resultats de structuration sur la journee du 12/05/2005. Deuxieme partie

Annexe F

Imprécision du guide des programmes

Cette annexe montre des exemples de guide de programmes prévisionnels, mis en correspondance avec la diffusion réelle (vérité terrain réalisée manuellement). Les tableaux F.1 et F.2 concernent la journée du 10/05/2005, et les tableaux F.3 et F.4 concernent la journée du 11/05/2005, toutes deux présentes dans le *corpus2*. Ces tableaux peuvent aussi être mis en correspondance avec les résultats de structuration automatique de l'annexe E.

Verite terrain			Guide des programmes		
15h28	15h42	Le renard	15h28	15h50	Le renard
15h48	16h36	Rex	15h50	16h40	Rex
16h36	16h37	Un livre			
16h43	16h57	Des chiffres et des lettres	16h40	16h45	Un livre
16h57	17h10	Des chiffres et des lettres	16h45	17h15	Des chiffres et des lettres
17h15	17h43	Tout vu, tout lu	17h15	18h00	Tout vu, tout lu
17h44	17h51	Tout vu, tout lu			
17h55	18h38	Urgences	18h0	18h50	Urgences
18h42	18h44	Cd'aujourd'hui			
18h49	19h39	On a tout essaye	18h50	19h50	On a tout essaye
19h41	19h42	Un jour un arbre			
19h46	19h48	MPGMC			
19h48	19h48	Pour Florence et Hussein			
19h49	19h55	Une journee de houf	19h50	20h00	Une journee de houf
19h55	19h56	Conso mag			
19h56	19h58	La meteo			
19h58	20h41	Journal	20h0	20h55	Journal
20h44	20h45	A parts egales			
20h45	20h47	La meteo			
20h51	20h52	Plus jamais comme ca			
20h58	22h31	Big mamma	20h55	22h45	Big mamma
22h39	22h44	Comme au cinema l'hebdo			
22h48	0h21	Une viree en enfer	22h45	0h25	Une viree en enfer
0h29	0h52	Journal de la nuit	0h25	0h50	Journal de la nuit
0h53	0h56	La meteo	0h50	1h55	Histoires courtes
1h00	1h02	Cd'aujourd'hui			
1h03	2h01	Histoires courtes	1h55	2h45	Chanter la vie
2h03	2h53	Chanter la vie	2h45	3h15	Trente millions d'amis
2h54	3h24	Trente millions d'amis	3h15	3h50	Les arts en liberte
3h24	3h59	Les arts en liberte	3h50	4h05	Journal de la nuit
3h59	4h23	Journal de la nuit	4h05	5h20	Faites entrer l'accuse
4h23	4h26	La meteo			
4h26	5h51	Faites entrer l'accuse	5h20	5h50	Journal
5h51	5h52	Un livre	5h50	5h55	Un livre

TAB. F.1 – Guide des programmes de la journee du 10/05/2005, mis en concordance avec la diffusion reelle. Premiere partie

Verite terrain			Guide des programmes		
5h54	6h19	Les z'amours	5h55	6h30	Les z'amours
6h19	6h25	Les z'amours			
6h25	6h26	Point route			
6h29	6h55	Telematin	6h30	8h30	Telematin
6h55	6h56	La meteo			
6h56	6h57	Point route			
7h0	7h28	Telematin			
7h32	7h51	Telematin			
7h54	8h11	Telematin			
8h14	8h23	Telematin			
8h24	8h25	La meteo			
8h25	8h27	Telematin			
8h27	8h28	Point route			
8h29	8h30	Un livre	8h30	8h35	Un livre
8h33	8h55	Des jours et des vies	8h35	9h00	Des jours et des vies
8h59	9h20	Amour, gloire et beaute	9h00	9h25	Amour, gloire et beaute
9h21	9h23	Cd'aujourd'hui			
9h26	9h52	Top of the pops	9h25	10h00	Top of the pops
9h57	10h18	KD2A	10h00	10h50	KD2A
10h22	10h46	KD2A			
10h49	10h53	Journal	10h50	11h00	Journal
10h57	11h20	Motus	11h00	11h35	Motus
11h21	11h28	Motus			
11h33	11h59	Les z'amours	11h35	12h10	Les z'amours
11h59	12h03	Les z'amours			
12h04	12h06	Cd'aujourd'hui			
12h11	12h29	La cible	12h10	13h00	La cible
12h30	12h47	La cible			
12h50	12h51	Le millionnaire			
12h55	12h56	Pour Florence et Hussein			
12h56	12h57	La meteo			
12h58	13h42	Journal	13h00	13h55	Journal
13h46	13h49	La meteo			
13h50	13h51	A parts egales			
13h53	14h41	Inspecteur Derrick	13h55	14h50	Inspecteur Derrick

TAB. F.2 – Guide des programmes de la journee du 10/05/2005, mis en concordance avec la diffusion réelle. Deuxieme partie

Verite terrain			Guide des programmes		
15h28	15h39	Le renard	15h28	15h55	Le renard
15h45	16h30	Rex	15h55	16h40	Rex
16h31	16h32	Un livre			
16h38	16h52	Des chiffres et des lettres	16h40	16h45	Un livre
16h53	17h05	Des chiffres et des lettres	16h45	17h20	Des chiffres et des lettres
17h12	17h40	Tout vu, tout lu	17h20	18h00	Tout vu, tout lu
17h40	17h48	Tout vu, tout lu			
17h52	18h35	Urgences	18h00	18h50	Urgences
18h38	18h40	Cd'aujourd'hui			
18h46	19h38	On a tout essaye	18h50	19h50	On a tout essaye
19h40	19h41	Un jour un arbre			
19h45	19h47	MPGMC			
19h48	19h54	Une journee de houf	19h50	20h0	Une journee de houf
19h55	19h55	Conso mag			
19h55	19h58	La meteo			
19h58	20h41	Journal	20h00	20h50	Journal
20h43	20h45	La meteo			
20h48	20h52	Loto	20h50	21h00	Loto
20h52	20h53	Plus jamais comme ca			
20h58	22h23	Une famille pas comme les autres	21h00	22h35	Une famille pas comme les autres
22h35	0h43	Ca se discute	22h35	0h50	Ca se discute
0h50	0h56	Talents Cannes 2005	0h50	1h00	Talents Cannes 2005
0h56	1h13	Journal de la nuit	1h00	1h25	Journal de la nuit
1h13	1h16	La meteo			
1h21	1h23	Cd'aujourd'hui			
1h24	3h03	Des mots de minuit	1h25	2h55	Des mots de minuit
3h03	3h48	La source de vie	2h55	3h55	Emissions religieuses
3h48	4h17	Emissions religieuses	3h55	4h10	Journal
4h18	4h35	Journal de la nuit	4h10	5h00	Lisbonne la bleue
4h35	4h37	La meteo			
4h38	5h25	Lisbonne la bleue	5h00	5h25	Outremers
5h25	5h51	Outremers	5h25	5h50	Journal
5h51	5h52	Un livre	5h50	5h55	Un livre
5h54	6h20	Les z'amours	5h55	6h30	Les z'amours
6h21	6h25	Les z'amours			

TAB. F.3 – Guide des programmes de la journee du 11/05/2005, mis en concordance avec la diffusion reelle. Premiere partie

Verite terrain			Guide des programmes		
6h25	6h26	Point route			
6h29	6h54	Telematin	6h30	8h35	Telematin
6h54	6h56	Telematin			
6h56	6h57	Point route			
6h59	7h25	Telematin			
7h25	7h27	Telematin			
7h27	7h27	Telematin			
7h31	7h55	Telematin			
7h58	8h12	Telematin			
8h15	8h30	Telematin			
8h30	8h31	Telematin			
8h31	8h37	Telematin	8h35	8h40	Un livre
8h41	8h42	Point route	8h40	9h05	Des jours et des vies
8h42	8h43	Un livre			
8h44	9h02	Des jours et des vies			
9h07	9h28	Amour, gloire et beaute	9h05	9h30	Amour, gloire et beaute
9h30	9h32	Cd'aujourd'hui	9h30	10h50	C'est au programme
9h36	10h9	C'est au programme			
10h13	10h45	C'est au programme			
10h49	10h53	Journal	10h50	11h00	Journal
10h58	11h21	Motus	11h00	11h35	Motus
11h22	11h29	Motus			
11h33	11h59	Les z'amours	11h35	12h10	Les z'amours
11h59	12h05	Les z'amours			
12h06	12h08	Cd'aujourd'hui			
12h12	12h30	La cible	12h10	12h50	La cible
12h31	12h46	La cible			
12h49	12h50	Loto	12h50	13h0	Loto
12h50	12h51	Le millionnaire			
12h51	12h52	Pour Florence et Hussein			
12h55	12h57	La meteo			
12h58	13h41	Journal	13h00	13h55	Journal
13h45	13h47	La meteo			
13h48	13h49	A parts egales			
13h51	14h38	Inspecteur Derrick	13h55	14h50	Inspecteur Derrick

TAB. F.4 – Guide des programmes de la journee du 11/05/2005, mis en concordance avec la diffusion reelle. Deuxieme partie

Annexe G

Navitex

Présentation

NAVITEX, pour Navigateur Avancé de VIdéo de l'équipe TEXmex, est un outil polyvalent, qui permet à la fois de créer des annotations manuelles de vidéos, de visualiser des résultats de segmentation automatique, et de naviguer agréablement dans une grande collection de vidéos. Cet outil est à l'origine un projet d'étudiants de l'IFSIC, et a été considérablement amélioré par Cédric Dufouil, le tout sous ma direction.

Navitex permet d'annoter une vidéo, et donc de créer les vérités terrain nécessaires à l'évaluation des algorithmes. Navitex travaille nativement dans le format TV-anytime [TVA02], qui a été construit pour décrire des programmes de télévision et, nous est, a priori, tout à fait adapté. Toutefois, le standard nous a montré ses limites en matière de description de contenus. S'il est parfaitement adapté pour être un format d'EPG, il devient beaucoup plus limité lorsqu'une description plus fine du flux est souhaitable, à des fins d'indexation. Le système des *Classification scheme* hérité de MPEG-7 [SS02] est particulièrement handicapant car il propose une liste arbitraire d'items à choisir, sans vraiment de possibilité d'extension. En particulier le descripteur *IntentionCS* (CS pour Classification Scheme) qui permet de définir le genre du programme, nous a paru lourd, mal adapté et incomplet, et a été modifié pour s'adapter à nos besoins.

TV-anytime est tout de même un standard relativement pratique, malgré les lourdeurs héritées de MPEG-7, et nous permet d'avoir une granularité temporelle fine, ainsi qu'un premier étage de structuration, par l'intermédiaire des *segmentation metadata*. Ce mécanisme permet de scinder un programme en plusieurs parties, par exemple, son générique de début, l'émission en elle-même, puis le générique de fin, ce qui permet de créer une description hiérarchique du flux.

Fonctionnalités

Segmentation et annotation

Navitex permet la création de vérité terrain de façon relativement conviviale. Si cette création reste toujours extrêmement pénible, elle est toutefois rendue beaucoup

facile par les outils fournis par Navitex. L'interface d'édition, présentée dans la figure G.1 permet une segmentation relativement aisée et rapide, et une description facilitée, en particulier, par le mécanisme des rediffusions, qui permet lorsqu'un programme est rediffusé, ce qui est très fréquent, de lui attribuer très simplement les métadonnées déjà écrites pour la version précédente. En l'absence de description, il est possible de naviguer

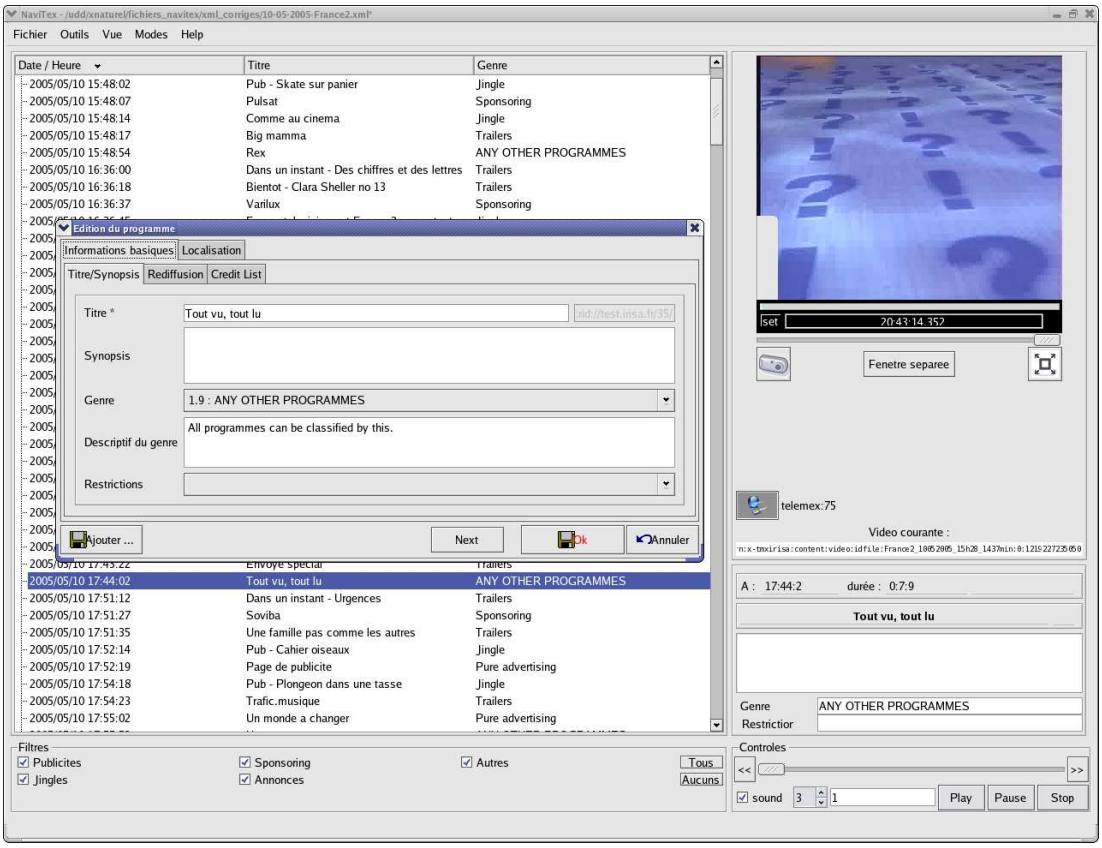

FIG. G.1 – L'interface d'édition de Navitex

à l'intérieur d'une vidéo de plusieurs façons : image par image, plan par plan[1], ou de façon plus classique, par l'intermédiaire d'un slider.

Un autre aspect est la capacité de Navitex à gérer des vidéos de grande taille[2], permettant un décodage rapide, un accès aléatoire précis et une navigation image par image. Navitex gère aussi relativement bien les descriptions de grande taille, mais la lourdeur d'un format basé XML transparait malgré tout pour des descriptions très lourdes, une segmentation en plan sur une vidéo de 24 heures par exemple.

[1] Segmentation en plans crée automatiquement, par l'algorithme de Manolis Delakis.

[2] Navitex a été abondamment utilisé sur des vidéos d'une durée de 24 heures encodées au format MPEG-2, soit un fichier de 40 Go

Visualisation de résultats d'étiquetage automatique

Navitex a été aussi concu pour pouvoir visualiser les résultats de segmentation et annotation automatique. Navitex peut tout à fait être utilisé pour visualiser les résultats d'une segmentation en plans par exemple, et éventuellement la modifier et l'enrichir grâce à l'interface d'annotation. L'intérêt principal et la motivation du développement de cet outil est toutefois de pouvoir visualiser les résultats d'un étiquetage automatique d'un flux de télévision.

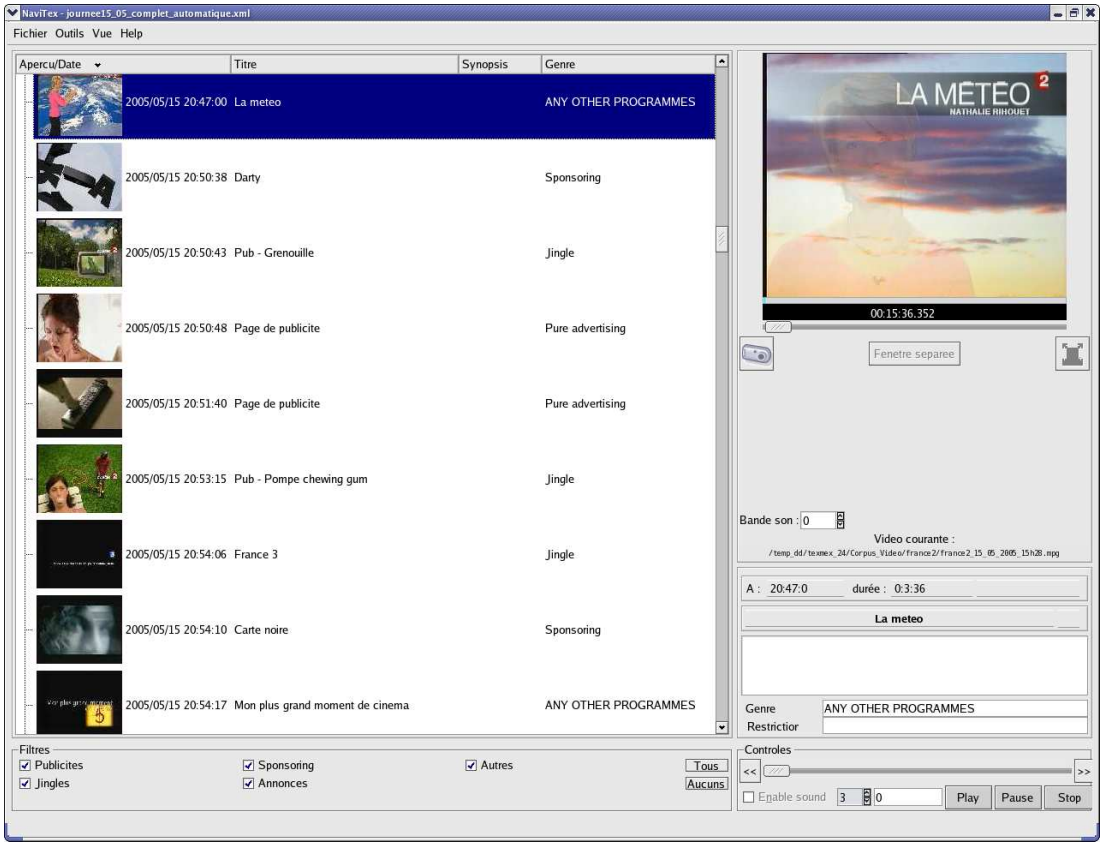

FIG. G.2 – L'interface de visualisation des résultats de Navitex

Les captures d'écran G.2 et G.3 montrent quelques exemples de la visualisation d'un étiquetage automatique. Il suffit de cliquer sur une imagette ou son étiquette associée pour jouer la séquence. Le fichier de description au format TV-anytime nécessaire à Navitex pour cette visualisation peut être soit directement crée par le programme producteur de la segmentation soit construit à partir d'un simple fichier texte au format beaucoup plus simple.

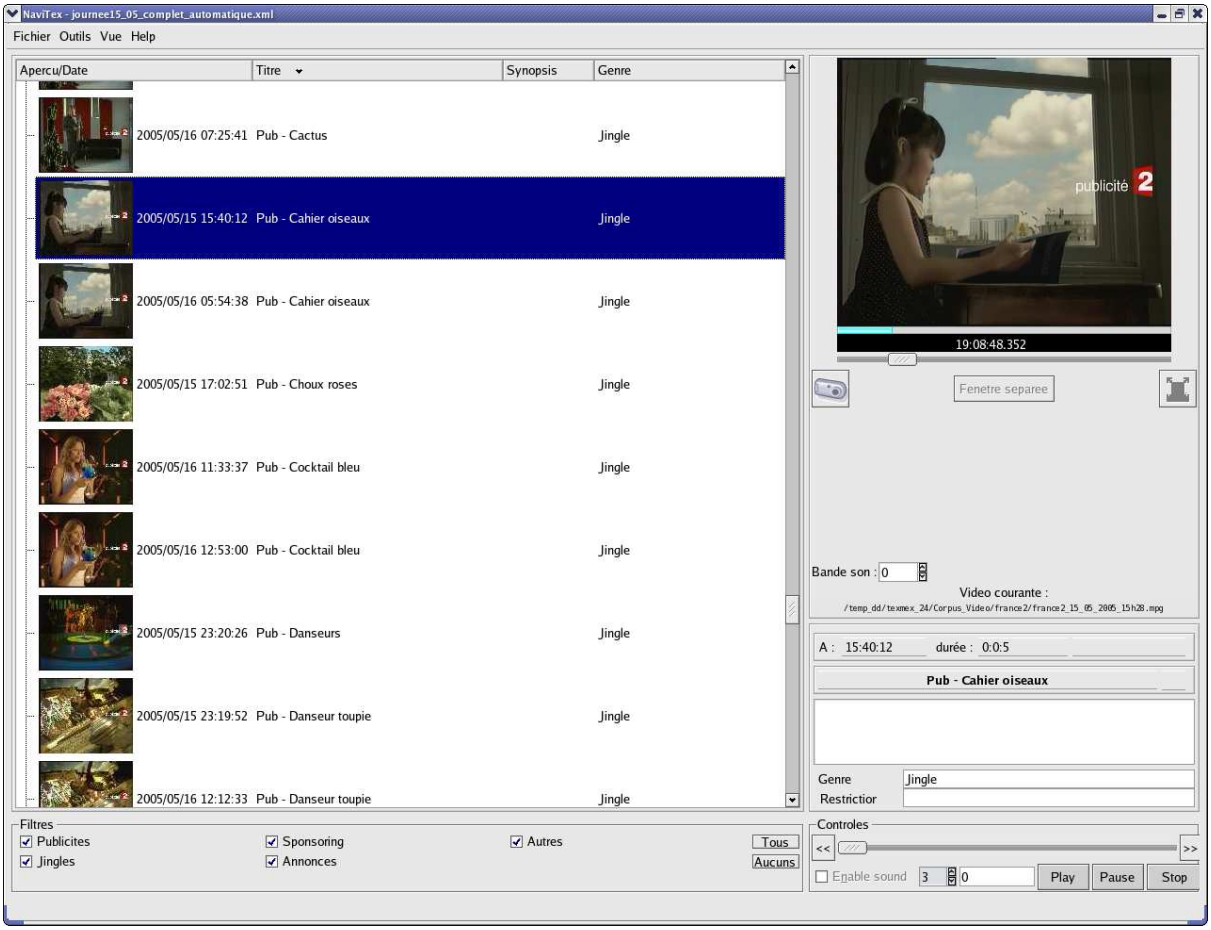

FIG. G.3 – Quelques résultats d'étiquetage automatique, triés ici par genre

Glossaire

CSA Conseil Supérieur de l'Audiovisuel
DCT Discrete Cosinus Transform
DTW Dynamic Time Warping
DVB Digital Video Broadcasting
EIT Event Information Table
EPG Electronic Program Guide
ESI Ensemble des segments inférés
EVR Ensemble de Vidéos de Référence
FERIA Framework pour l'Expérimentation et la Réalisation Industrielle d'Applications multimédias
FFTW Fastest Fourier Transform in the West
HMM Hidden Markov Model
INA Institut National de l'Audiovisuel
IP Inter-programme
KLT Karhunen-Loève Transform
MPEG Moving Picture Expert Group
NED Normalized Edit Distance
OCR Optical Character Recognition
PDC Program Delivery Control
SVM Support Vector Machine
TAL Traitement automatique des langues
TEB Taux d'Erreur Bit
VPS Video Programming System

Bibliographie

[AFAT04] Alberto Albiol, Maria José Ch. Fulla, Antonio Albiol, and Luis Torres. Commercials detection using hmm's. In *Image Analysis for Multimedia Interactive Services, Wiamis'2004, Lisboa, Portugal*, April 2004.

[AJL95] P. Aigrain, P. Joly, and V. Longueville. Medium knowledge-based macro-segmentation of video into sequences. In *Proc. IJCAI Workshop on Intelligent Multimedia Information Retrieval, Montréal*, 1995.

[ALK99] Donald A. Adjeroh, M. C. Lee, and Irwin King. A distance measure for video sequences. *Comput. Vis. Image Underst.*, 75(1-2) :25–45, 1999.

[All83] James F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11) :832–843, 1983.

[AS95a] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In Philip S. Yu and Arbee S. P. Chen, editors, *Eleventh International Conference on Data Engineering*, pages 3–14, Taipei, Taiwan, 1995. IEEE Computer Society Press.

[AS95b] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In Philip S. Yu and Arbee S. P. Chen, editors, *Eleventh International Conference on Data Engineering*, pages 3–14, Taipei, Taiwan, 1995. IEEE Computer Society Press.

[AVBD05] Brett Adams, Svetha Venketesh, Hung Bui, and Chitra Dorai. A probabilistic framework for extracting narrative act boundaries and semantics in motion pictures. *Multimedia Tools and Applications*, 27 :195 – 213, November 2005.

[BBH03] J. Barr, B. Bradley, and B.T. Hannigan. Using digital watermarks with image signatures to mitigate the threat of the copy attack. In *ICASSP*, 2003.

[BBP01] M. Bertini, A. Del Bimbo, and P. Pala. Content-based indexing and retrieval of tv news. *Pattern Recogn. Lett.*, 22(5) :503–516, 2001.

[Ber04] Sid-Ahmed Berrani. *Recherche approximative de plus proches voisins avec contrôle probabiliste de la précision ; application à la recherche d'images par le contenu.* Thèse de doctorat, Université de Rennes1, February 2004.

[BGG99] P. Bouthemy, M. Gelgon, and F. Ganansia. A unified approach to shot change detection and camera motion characterization. *IEEE Trans. on Circuits and Systems for Video Technology*, 9(7) :1030–1044, October 1999.

[BW98] J. Boreczky and L. Wilcox. A hidden markov model framework for video segmentation using audio an image features. In *Proc. IEEE ICASSP*, Seattle, 1998.

[CA04] Sabine Carbonnel and Eric Anquetil. Lexicon organization and string edit distance learning for lexical post-processing in handwriting recognition. In *IWFHR '04 : Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition (IWFHR'04)*, pages 462–467, Washington, DC, USA, 2004. IEEE Computer Society.

[Car04] Jean Carrive. Feria : Framework pour l'expérimentation et la réalisation industrielle d'applications multimédias. In *RIAM*, page 10, Rennes, France, 2004.

[CBF06] Michele Covell, Shumeet Baluja, and Michael Fink. Advertisement detection and replacement using acoustic and visual repetition. In *MMSP'06, IEEE 8th workshop on Multimedia Signal Procesing*, October 2006.

[CF01] M. Cooper and J. Foote. Scene boundary detection via video self-similarity analysis. In *Proc. IEEE Intl. Conf. on Image Processing*, pages 378–381, 2001.

[CH89] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. In *Proceedings of the 27th annual meeting on Association for Computational Linguistics*, pages 76–83, Morristown, NJ, USA, 1989. Association for Computational Linguistics.

[Che06] Guillaume Chesnel. Suivi et reconnaissance de texte pour l'indexation de vidéos de télévision. Technical report, (Rapport de Master). IFSIC - Université de Rennes1, Septembre 2006.

[CM02] A. Cornuéjols and L. Miclet. *Apprentissage artificiel. Concepts et algorithmes.* Eyrolles, 2002.

[CN05] S.-C. Cheung and T. Nguyen. Mining arbitrary-length repeated patterns in television broadcast. In *IEEE International Conference on Image Processing, ICIP 2005*, pages 181–184, September 2005.

[CS04] B. Coskun and B. Sankur. Robust video hash extraction. In *EUSIPCO : European Conf. On Signal Processing*, Vienna, 2004.

[CYC+05] Jun-Cheng Chen, Jen-Hao Yeh, Wei-Ta Chu, Jin-Hau Kuo, and Ja-Ling Wu. Improvement of commercial boundary detection using audiovisual features. In *PCM*, pages 776–786, 2005.

[DCH04] P. Duygulu, Ming-Yu Chen, and Alex Hauptmann. Comparison and combination of two novel commercial detection methods. In *ICME*, Taipei, Taiwan, june 2004.

[DJN+02] N. Dimitrova, S. Jeannin, J. Nesvadba, T. McGee, L. Agnihotri, and G. Mekenkamp. Real-time commercial detection using mpeg features. In *Proc. 9th Int. Conf. On Information Processing and Management of Uncertainty in knowledge-based systems*, pages 481–486, 2002.

[Dom00] Jean-Claude Domenget. La multiplicité des temps télévisuels : de la production à la réception. In *Les temps médiatiques, temporalistes*, number 42, 2000.

[DVB03] Etsi, es201-812, digital video broadcasting, multimedia home platform specifications 1.0.3, 2003.

[EBU93] EBU. Ets 300 231, television systems ; specification of the domestic video programme delivery control system (pdc), 1993.

[EM99] Stefan Eickeler and Stefan Müller. Content-Based Video Indexing of TV Broadcast News Using Hidden Markov Models. In *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 2997–3000, Phoenix, 1999.

[FBGS04] B. Fauvet, P. Bouthemy, P. Gros, and F. Spindler. A geometrical key-frame selection method exploiting dominant motion estimation in video. In *CVIR'04*, Dublin, Ireland, July 2004.

[FC03] J. Foote and M. Cooper. Media segmentation using selfsimilarity decomposition. In *Proc. SPIE Storage and Retrieval for Multimedia Databases*, volume 5021, pages 167–75, 2003.

[FD81] D. Freedman and P. Diaconis. On the histogram as a density estimator : L2 theory. *Zeitschrift fur Wahrscheinlichkeitstheorie und verwandte Gebiete*, 57 :453–476, 1981.

[FG00] Jiri Fridrich and Miroslav Goljan. Robust hash functions for digital watermarking. In *Proceedings. International Conference on Information technology*, pages 178–183, March 2000.

[FGS90] Christine Froidevaux, Marie-Claude Gaudel, and Michèle Soria. *Types de données et algorithmes*. Ediscience international, 1990.

[FJ05] Matteo Frigo and Steven G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2) :216–231, 2005. special issue on "Program Generation, Optimization, and Platform Adaptation".

[GDPL00] J-C. Bassano Gaël Dias, Sylvie Guillore and José Gabriel Pereira-Lopes. Extraction automatique d'unités lexicales complexes : Un enjeu fondamental pour la recherche documentaire. *Traitement automatique des langues pour la recherche d'information*, 1(2) :447–473, 2000.

[GS06a] John M. Gauch and Abhishek Shivadas. Finding and identifying unknown commercials using repeated video sequence detection. *Comput. Vis. Image Underst.*, 103(1) :80–88, 2006.

[GS06b] John M. Gauch and Abhishek Shivadas. Real-time commercial recognition using color moments and hashing. In *ACM SIGMM Int. Workshop on Multimedia Information Retrieval*, Santa Barbara, CA, USA, 2006.

[Hai05] Siba Haidar. *Comparaison des documents audiovisuels par matrice de similarité* . Thèse de doctorat, Université Paul Sabatier, Toulouse, France, septembre 2005.

[Han02] Alan Hanjalic. Shot-boundary detection : unraveled and resolved ? *IEEE Trans. Circuits Syst. Video Techn.*, 12(2) :90–105, 2002.

[HB00] A. Hampapur and R. Bolle. Feature based indexing for media tracking. In *ICME*, 2000.

[HCA01] Hadi Harb, Liming Chen, and Jean-Yves Auloge. Speech/music/silence and gender detection algorithm. In *Proceedings of the 7th International Conference on Distributed Multimedia Systems (DMS 01)*, 2001.

[Her05] Cormac Herley. Accurate repeat finding and object skipping using fingerprints. In *MULTIMEDIA '05 : Proceedings of the 13th annual ACM international conference on Multimedia*, pages 656–665, New York, NY, USA, 2005. ACM Press.

[HFBYL92] D. Harman., E. Fox, R.A. Baeza-Yates, and W. Lee. *Data Structures and Algorithms : Inverted Files.* Prentice Hall, 1992.

[HGN00] Jochen Hipp, Ulrich Güntzer, and Gholamreza Nakhaeizadeh. Algorithms for association rule mining — a general survey and comparison. *SIGKDD Explorations*, 2(1) :58–64, July 2000.

[Hop01] F. Hoppner. Learning temporal rules from state sequences. In *Proc. of the IJCAI'01 Workshop on Learning from Temporal and Spatial Data*, pages 25–31, Seattle, USA, 2001.

[HZ03] Timothy C. Hoad and Justin Zobel. Fast video matching with signature alignment. In *MIR '03 : Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pages 262–269, New York, NY, USA, 2003. ACM Press.

[IMK03] Ichiro Ide, Hiroshi Mo, and Norio Katayama. Threading news video topics. In *MIR '03 : Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pages 239–246, New York, NY, USA, 2003. ACM Press.

[Jai89] Anil K. Jain. *Fundamentals of digital image processing.* Prentice hall information and system sciences series, 1989.

[JBR04] Sung Ho Jin, Tae Meon Bae, and Yong Man Ro. Automatic video genre detection for content-based authoring. In *PCM (1)*, pages 335–343, 2004.

[JL01] R. S. Jasinschi and J. Louie. Automatic tv program genre classification based on audio patterns. *Euromicro*, 00 :0370, 2001.

[Jol05] Alexis Joly. *recherche par similarité statistique dans une grande base de signatures locales pour l'identification rapide d'extraits vidéos.* Thèse de doctorat, Université de la Rochelle, 2005.

[KHS05] Yan Ke, Derek Hoiem, and Rahul Sukthankar. Computer vision for music identification. In *Computer Vision and Pattern Recognition*, June 2005.

[Kij03] Ewa Kijak. *Structuration multimodale des vidéos de sports par modèles stochastiques*. Thèse de doctorat, Université de Rennes 1, December 2003.

[KKM03] Kunio Kashino, Takayuki Kurozumi, and Hiroshi Murase. A quick search method for audio and video signals based on histogram pruning. *IEEE Transactions on Multimedia*, 5 :348–357, 2003.

[KP03] Woong Hee Kim and Il Hwan Park. Image authentication using relationships of vectors. *STEG'03, Pacific Rim Workshop on Digital Steganography 2003*, July 2003.

[LCM03] Frédéric Lefèbvre, Jacek Czyz, and Benoit M. Macq. A robust soft hash algorithm for digital image signature. In *ICIP (2)*, pages 495–498, 2003.

[Lev65] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Doklady Akademii Nauk SSSR*, 163(4) :845–848, 1965.

[LHÁTJA06] Herwig Lejsek, Fridrik H. Ásmundsson, Björn Thór-Jónsson, and Laurent Amsaleg. Blazingly fast image copyright enforcement. In *14th ACM International Conference on Multimedia, demonstrations*, Santa Barbara, CA, USA, October 2006.

[Lie01] Rainer Lienhart. Reliable transition detection in videos : A survey and practitioner's guide. *Int. J. Image Graphics*, 1(3) :469–486, 2001.

[LKE97] R. Lienhart, C. Kuhmunch, and W. Effelsberg. On the detection and recognition of television commercials. In *International Conference on Multimedia Computing and Systems*, pages 509–516, 1997.

[LL00] Chun-Shien Lu and Hong-Yuan Mark Liao. Structural digital signature for image authentication : an incidental distortion resistant scheme. In *MULTIMEDIA '00 : Proceedings of the 2000 ACM workshops on Multimedia*, pages 115–118. ACM Press, 2000.

[LLXT05] Liuhong Liang, Hong Lu, Xiangyang Xue, and Yap-Peng Tan. Program segmentation for tv videos. In *ISCAS, IEEE International Symposium on Circuits and Systems*, volume 2, pages 1549–1552, 2005.

[LQZ04] Tie-Yan Liu, Tao Qin, and HongJiang Zhang. Time-constraint boost for tv commercials detection. In *ICIP*, pages 1617–1620, 2004.

[Mar] Pascal Marie. Content protection and rights management, specificities of video watermarking. http ://www.broadcastpapers.com/.

[MB03] D. Marquis and S. Bres. Suivi et amélioration de textes issus de génériques vidéos. In *Journées d'Etudes et d'Echanges 'Compression et Représentation des Signaux Audiovisuels'*, pages 179–182, 2003.

[McK03] David McKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[MD99] T. McGee and N. Dimitrova. Parsing tv program structures for identification and removal of non-story segments. In *in SPIE Conf. on Storage and Retrieval for Image and Video Databases*, 1999.

[Mer06] Maya Merheb. Macrosegmentation par matrice de similarité. Technical report, (Rapport de Master). IRIT, Université Paul Sabatier, Université Libanaise., Septembre 2006.

[MLM04] Antonio Mucedero, Rosa Lancini, and Francesco Mapelli. A novel hashing algorithm for video sequences. In *ICIP*, pages 2239–2242, Singapore, 2004.

[MV93] A. Marzal and E. Vidal. Computation of normalized edit distance and applications. *IEEE Trans. PAMI*, 15(9) :926–932, 1993.

[Naf94] J. Nafeh. "method and apparatus for classifying patterns of television program and commercials based on discerning of broadcast audio and video signals. US patent 5,343,251, august 1994.

[Neu75] D.L. Neuhoff. The viterbi algorithm as an aid in text recognition. *IEEE Trans. Inform. Theory*, 21(2) :222–226, March 1975.

[OGPG03] J-M. Odobez, D. Gatica-Perez, and M. Guillemot. Video Shot Clustering using Spectral Methods. In *3rd Workshop on Content-Based Multimedia Indexing (CBMI)*, Rennes, France, septembre 2003.

[OKH02] Job Oostveen, Ton Kalker, and Jaap Haitsma. Feature extraction and a database strategy for video fingerprinting. In *VISUAL '02 : Proceedings of the 5th International Conference on Recent Advances in Visual Information Systems*, pages 117–128. Springer-Verlag, 2002.

[PBKD05] F. Pitié, S.-A. Berrani, A. Kokaram, and R. Dahyot. Off-line multiple object tracking using candidate selection and the viterbi algorithm. *Proc. of the IEEE International Conference on Image Processing*, September 2005.

[PBY04] G. Piriou, P. Bouthemy, and J-F. Yao. Extraction of semantic dynamic content from videos with probabilistic motion models. In *Proc. Eur. Conf. Computer Vision (ECCV'04*, Prague, Czech Republic, May 2004.

[PGGM04] Kok Meng Pua, John M. Gauch, Susan E. Gauch, and Jedrzej Z. Miadowicz. Real time repeated video sequence identification. *Comput. Vis. Image Underst.*, 93(3) :310–327, 2004.

[Pin04] Julien Pinquier. *Indexation sonore : recherche de composantes primaires pour une structuration audiovisuelle.* Thèse de doctorat, Université Paul Sabatier, Toulouse, France, décembre 2004.

[Pol07] Jean-Philippe Poli. *Prédiction de programmes de télévision pour la structuration vidéo automatique.* Thèse de doctorat, INA/Université Paul Cézanne, Marseille, 2007.

[RH04] Seungmin Rho and Eenjun Hwang. Video scene determination using audiovisual data analysis. In *ICDCSW '04 : Proceedings of the 24th International Conference on Distributed Computing Systems Workshops - W7 : EC (ICDCSW'04)*, pages 124–129, Washington, DC, USA, 2004. IEEE Computer Society.

[RHM99] Yong Rui, Thomas S. Huang, and Sharad Mehrotra. Constructing table-of-content for videos. *Multimedia Systems*, 7(5) :359–368, 1999.

[RY90] K.R Rao and P Yip. *Discrete Cosine Transform*. Acadamec Press, 1990.

[RY98] Eric Sven Ristad and Peter N. Yianilos. Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5) :522–532, 1998.

[Sat04] O. Satterwhite, B.; Marques. Automatic detection of tv commercials. In *IEEE Potentials*, volume 23, pages 9 – 12, April 2004.

[SB04] Izhak Shafran and William Byrne. Task-specific minimum bayes-risk decoding using learned edit distance. In *Proc. of the International Conference on Spoken Language Processing*, 2004.

[SBV02] Juan María Sánchez, Xavier Binefa, and Jordi Vitrià. Shot partitioning based recognition of tv commercials. *Multimedia Tools Appl.*, 18(3) :233–247, 2002.

[SC78] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 26(1) :43–49, 1978.

[sdl] Conseil supérieur de l'audiovisuel. Publicité, parrainage et téléachat à la télévision et à la radio, http ://www.csa.fr/.

[Sig04] François Signol. Vers une inférence de motifs non-supervisée. Technical report, (Rapport de DEA). ENSEIRB - Université de Bordeaux I, août 2004.

[SJ72] K. Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1) :11–21, 1972.

[Sme05] Alan Smeaton. Shot boundary task overview. In *TRECVID 2005 Workshop*, November 2005.

[SMOM02] D. Sadlier, S. Marlow, N. OConnor, and N. Murphy. Automatic tv advertisement detection from mpeg bitstream. *Journal of the Patt. Rec. Society*, 35 :2–15, 2002.

[SS02] Phillipe Salembier and Thomas Sikora. *Introduction to MPEG-7 : Multimedia Content Description Interface*. John Wiley & Sons, Inc., New York, NY, USA, 2002.

[TDV00] Ba Tu Truong, Chitra Dorai, and Svetha Venkatesh. New enhancements to cut, fade, and dissolve detection processes in video segmentation. In *MULTIMEDIA '00 : Proceedings of the eighth ACM international conference on Multimedia*, pages 219–227, New York, NY, USA, 2000. ACM Press.

[TKR99] Y.-P. Tan, S. R. Kulkarni, and P. J. Ramadge. A framework for measuring video similarity and its application to video query by example. In *Proceedings ICIP 99*, 1999.

[TPBD03] Cüneyt M. Taskiran, Ilya Pollak, Charles A. Bouman, and Edward J. Delp. Stochastic models of video structure for program genre detection. In *VLBV*, pages 84–92, 2003.

[TVA02] Tv-anytime forum, specification series on metadata. parta : Metadata schemas. http ://www.tv-anytime.org/, 2002.

[Ven02] E. Veneau. *Macro-segmentation multi-critère et classification de séquences par le contenu dynamique pour l'indexation vidéo.* Thèse de doctorat, Université de Rennes I, Mention Traitement du Signal, February 2002.

[WBSS04] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment : From error visibility to structural similarity. *IEEE Trans. Image Processing*, 13, 2004.

[WC06] Hsu Winston and Shih-Fu Chang. Topic tracking across broadcast news videos with visual duplicates and semantic concepts. In *ICIP*, Atlanta, October 2006.

[WHHF99] Xiaodong Wen, Theodore D. Huffmire, Helen H. Hu, and Adam Finkelstein. Wavelet-based video indexing and querying. *Multimedia Systems*, 7(5) :350–358, 1999.

[WJ03] C. Wolf and J.M. Jolion. Extraction and recognition of artificial text in multimedia documents. In *Pattern Analysis and Applications*, 2003.

[XSH05] Hong-Jiang Zhang Xian-Sheng Hua, Lie Lu. Robust learning-based tv commercial detection. In *ICME*, july 2005.

[YDTX04] Junsong Yuan, Ling-Yu Duan, Qi Tian, and Changsheng Xu. Fast and robust short video clip search using an index structure. In *MIR '04 : Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pages 61–68. ACM Press, 2004.

[YR04] Zhen Yao and Nasir Rajpoot. Radon/ridgelet signature for image authentication. In *ICIP*, pages 43–46, Singapore, 2004.

[ZH06] Justin Zobel and Timothy C. Hoad. Detection of video sequences using compact signatures. *ACM Trans. Inf. Syst.*, 24(1) :1–50, 2006.

[ZZC96] Di Zhong, HongJiang Zhang, and Shih-Fu Chang. Clustering methods for video browsing and annotation. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 239–246, 1996.

Publications de l'auteur

Journal

Xavier Naturel, Patrick Gros. Detecting Repeats for Video Structuring. Multimedia Tools and Application, À paraître.

Workshops et Symposium

- Xavier Naturel, Guillaume Gravier, P. Gros. Fast Structuring of Large Television Streams using Program Guides. 4th International Workshop on Adaptive Multimedia Retrieval (AMR), Volume 4398, pages 223-232, Geneva, Switzerland, Juillet 2006.
- Xavier Naturel, Patrick Gros. A Fast Shot Matching Strategy for detecting duplicate sequences in a television stream. CVDB'05 : Proceedings of the 2nd ACM SIGMOD International Workshop on Computer Vision meets DataBases, pages 21-27, Baltimore, USA, Juin 2005.

Conférences Nationales

Xavier Naturel, Guillaume Gravier, Patrick Gros. Étiquetage Automatique de Programmes de Télévision. CORESA'05 Compression et représentation des signaux audiovisuels, Rennes, France, Novembre 2005.

Rapports de recherche

Xavier Naturel, Patrick Gros. Detecting Repeats for Video Structuring. Rapport de Recherche IRISA, No 1790, Mars 2006.

Résumé

La structuration automatique de flux de télévision est un nouveau sujet de recherche, dont l'apparition est liée à l'augmentation de volume des archives de vidéos numériques de télévision. Cette thèse propose une chaine complète de structuration, qui permet de segmenter et d'étiqueter automatiquement un flux télévisé. Les travaux présentés se divisent en quatre parties : la définition d'outils, la segmentation, l'étiquetage, et la mise à jour.

Un flux de télévision est intrinsèquement répétitif. L'une des idées directrices de la thèse est de considérer les répétitions comme une aide essentielle pour la structuration, en particulier pour réaliser la distinction entre les programmes et les inter-programmes. Une méthode rapide de détection des répétitions dans des flux vidéos est proposée, permettant de gérer d'importants volumes vidéos, à partir d'une base de vidéos de référence, étiquetée manuellement. Grâce à cet outil, ainsi qu'à la détection des séparations entre publicités, une segmentation en programmes/inter-programmes est réalisée. Les segments sont alors étiquetés à partir du guide des programmes, en réalisant un alignement global par dynamic time warping. Enfin, une étape de mise à jour permet de réduire la dépendance à une base de référence manuelle, ainsi que de réduire la baisse de qualité des résultats de structuration au cours du temps.

Mots-clés : Structuration vidéo, indexation vidéo, macro-segmentation, hachage perceptuel, multimédia, télévision.

Abstract

Automatic television structuring is a new research topic. Its development is motivated by the growth of digital television archives. This thesis proposes a framework for television structuring, which allows to automatically segment and label television streams. The work is composed of four parts : tools definition, stream segmentation, labeling and update.

Television streams are inherently repetitive. One of the main ideas of this thesis is to take advantage of the repetitions to help structuring the stream. Repetitions are especially useful to detect non-programs, because of their repetitive nature. A fast method for detecting repetitions is proposed, which can deal with large amount of video, and which uses a manually labeled reference video dataset. Together with the detection of commercial separations, the repetition detection is used as an input to the segmentation process, resulting in a partition of the stream into programs and non-programs. The program guide is then used to label each segment, using a dynamic time warping algorithm. Eventually, a method for updating the reference video dataset is proposed, which allows to be less dependant from a manually labeled dataset. This update step also helps to maintain a constant structuring quality over time.

Keywords : video structuring, video indexing, video fingerprinting, perceptual hashing, multimedia, television.